Smart mining system with crystal classification of ores and industrial management

MAJOR PROJECT REPORT

Submitted by

MOHAMED RIZWAN [RA2011029010014]
YASHU YOUWRAJ [RA2011029010062]

Under the Guidance of

Dr.K.KALAISELVI

Associate Professor, Department of Networking and Communications

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
with specialization in (SPECIALIZATION NAME)

DEPARTMENT OF NETWORKING AND COMMUNICATIONS
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203

MAY 2024

Department of Networking and Communications SRM Institute of Science and Technology

SRM Institute of Science & Technology
Own Work* Declaration Form

Degree/ Course	**:**	B.Tech in Computer Science and Engineering with a specialization in Computer Networking
Student Name	**:**	Mohamed Rizwan, Yashu Youwraj
Registration Number	**:**	RA2011029010014, RA2011029010062
Title of Work	**:**	Smart mining system with crystal classification of ores and industrial management

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is my / our own except where indicated and that We have met the following conditions:

- Referenced/listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not making any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

We understand that any false claim for this work will be penalized by the University policies and regulations.

DECLARATION:	
We are aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is our work, except where indicated by referring, and that I have followed the good academic practices noted above.	
Mohamed Rizwan K (RA2011029010014)	Yashu Youwraj (RA2011029010062)

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V. Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr Annapurani K**, Professor and Head, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinator, **Dr. G. Suseela**, Associate Professor, Panel Head, **Dr. K. Deepa Tilak,** Associate Professor and members, **Dr. M. Manickam**, Assistant Professor, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr K. Kalaiselvi,** Associate Professor, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my / our guide, **Dr K. Kalaiselvi**, Associate Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for providing me/us with an opportunity to pursue our project under his/her mentorship. He/She provided me/us with the freedom and support to explore the research topics of my/our interest. His/Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Networking and Communications, Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, I/we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement

Mohamed Rizwan[RA2011029010014]

Yashu Youwraj[RA2011029010062]

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSP109L/18CSP111L project report [18CSP112L Internship report] titled "**Smart mining system with crystal classification of ores and industrial management** " is the bonafide work of "**Mr.Mohamed Rizwan [Reg. No.: RA2011029010014]** and **Mr. Yashu Youwraj [Reg. No.RA2011029010062]** who carried out the project work[internship] under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr.K.KALAISELVI
SUPERVISOR
Associate Professor
Department of
Networking and
Communications

Dr. K. ANNAPURANI
HEAD OF THE DEPARTMENT
DEPARTMENT OF NETWORKING AND COMMUNICATION

Internal Examiner

External Examiner

TABLE OF CONTENTS

ABSTRACT

To guarantee a dependable supply of raw materials that are essential for modern living and the transition towards environmentally friendly technologies, mineral exploration is the most important thing that can be done. During the mining process, expensive procedures are carried out to locate areas in the crust of the Earth that contain naturally occurring mineral concentrations. It is possible that the costs associated with these operations could be significantly reduced by combining techniques that involve artificial intelligence and remote sensing. Specifically, it is designed to identify possible areas to extract the desired composition of minerals, and it presents a powerful and intelligent model for mineral exploration. This model is presented here. A sophisticated deep learning process that makes use of a random forest algorithm to analyse the dataset is incorporated into our approach, which also incorporates cutting-edge developments in artificial intelligence and remote sensing. Discovering the different kinds of ores that can be extracted from the minerals that are provided is the primary objective. Beyond simply identifying things, this method can be used for a wider range of purposes. The type of soil that is used to extract the ores can also be determined with the help of this tool. In addition to applying to a wide variety of ore deposit models and datasets, this versatile method is not limited to a single ore source. A significant step forward in the field of mineral exploration is the application of deep learning to the process of analysing data pertaining to ores. In addition to making a significant contribution to the sustainable acquisition of raw materials and the worldwide shift towards environmentally friendly technologies, it has the potential to improve the efficiency, precision, and cost-effectiveness of identifying areas that contain deposits of abundant minerals.

LIST OF FIGURES

ABBREVIATIONS

3D	Three Dimensional
PC	Personal Computer
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
AI	Artificial Intelligence
ML	Machine Learning
C3D	Convolutional 3 Dimensional
LSTM	Long Short-Term Memory
ConvLTSM	Convolutional Long Short-term Memory
JTSM	Joint Time series modelling
FLOPS	Floating-point operations second
SiLU	Sigmoid Linear Units
I.E	id est
E.G	exempli gratia
Eq	Equation
GPU	Graphics Processing Unit
RAM	Random-Access Memory
OpenCV	Open source Computer Vision library
DL	Deep Learning

CHAPTER 1

INTRODUCTION

1.1 Introduction

The mining sector is crucial in balancing technological advancements and environmental responsibility by promoting sustainable resource utilisation and optimising industrial processes. The conventional mining models are undergoing changes, motivated by the need to optimise productivity, minimise ecological consequences, and establish an accountable supply chain for crucial raw materials. The incorporation of advanced technologies, specifically artificial intelligence and data analytics, has led to the emergence of the Smart Mining System concept. This paper explores the design and execution of an advanced mining system that goes beyond traditional mining methods. Our focus goes beyond simple resource extraction and includes a comprehensive approach that incorporates Crystal Classification of Ores and advanced Industrial Management techniques. This system aims to revolutionise the mining industry by utilising cutting-edge technologies. It will address significant challenges and contribute to a more sustainable and intelligent mining sector. The primary goal of the Smart Mining System with Crystal Classification of Ores and Industrial Management is to improve the effectiveness, durability, and output of mining activities. Create a sophisticated crystal classification system that incorporates cutting-edge technologies, such as machine learning, to precisely and reliably identify and categorise mineral ores. The system must be able to effectively manage a wide range of ore compositions and geological variations. Smart mining is crucial for life and valuable for future forecasting. Next, we will identify the practical application and the project's specific range.

Classification of ores based on crystal structure: Efficient extraction processes rely on the crucial identification and classification of minerals present in ores. The Smart Mining System we have developed utilises sophisticated crystal classification algorithms, frequently driven by machine learning, to accurately analyse and classify different types of ores.

This not only improves the efficiency of extracting resources but also enables the optimal utilisation of resources.

The Smart Mining System incorporates advanced Industrial Management techniques in addition to resource extraction. This entails the continuous monitoring of operations, analysis of data, and optimisation of processes to make mining operations more efficient. Effective industrial management facilitates the smooth integration of different elements within the mining ecosystem, resulting in enhanced efficiency and decreased operational expenses.

1.2.Purpose

A smart mining system with crystal classification of ores and industrial management serves several purposes, all aimed at optimizing the mining process and improving overall efficiency and productivity in the mining industry. Here are some key purposes and benefits:

Optimized Resource Extraction: The system allows for the precise identification and classification of ores based on their crystal structures. This enables miners to target specific mineral deposits more accurately, reducing wastage and increasing the overall yield of valuable minerals.

Cost Reduction: By streamlining the mining process and reducing inefficiencies, the smart mining system helps to lower operational costs. This includes reduced energy consumption, optimized equipment usage, and minimized downtime.

Improved Safety: With advanced technologies such as sensors, drones, and AI-powered analytics, the system enhances safety measures in mining operations. It can detect potential hazards in real time, alerting workers and preventing accidents before they occur.

Data-Driven Decision-Making: The system collects and analyzes vast amounts of data from various sources, including sensors, equipment, and geological surveys. This data can be used to make informed decisions about resource allocation, equipment maintenance, and operational planning, leading to better outcomes and increased profitability.

Environmental Sustainability: By optimizing resource utilization and reducing waste, the smart mining system promotes environmentally sustainable mining practices. It helps

minimize the ecological footprint of mining operations, mitigating negative impacts on the environment.

Enhanced Industrial Management: The system provides tools for comprehensive industrial management, including inventory tracking, supply chain optimization, and workforce management. This improves overall efficiency and coordination across different aspects of the mining operation.

Remote Monitoring and Control: With remote monitoring capabilities, operators can oversee mining activities from a central control center, even across multiple sites. This enables real-time monitoring of operations and allows for immediate adjustments to be made in response to changing conditions.

Overall, a smart mining system with crystal classification of ores and industrial management offers a holistic approach to modernizing the mining industry, leading to increased productivity, safety, and sustainability.

1.3.Scope

Smart mining is the most important of life and useful for future prediction. Then, identified useful applications and the scope the project. The project explores as,

Automated Ore Identification: Implement image recognition or spectroscopy techniques to identify and classify different crystals and ores. Use machine learning algorithms to improve the accuracy of ore classification over time.

Smart Mining Equipment: Integrate sensors and IoT devices into mining equipment for real-time data collection. Implement automation for drilling, blasting, and material transportation.

Industrial Management System: Develop a centralized system for monitoring and controlling various aspects of mining operations. Include modules for inventory management, equipment maintenance, and workforce scheduling.

Energy Efficiency: Implement energy-efficient practices by optimizing equipment usage and scheduling. Integrate renewable energy sources where feasible.
Data Analytics: Utilize data analytics to derive insights from the vast data generated during mining operations.Predictive maintenance to reduce downtime and increase equipment lifespan.

Safety and Compliance: Include safety features such as real-time monitoring of environmental conditions and worker health. Ensure compliance with industry regulations and standards.

1.4..Objectives

The Smart Mining System with Crystal Classification of Ores and Industrial Management aims to achieve several key objectives to enhance the efficiency, sustainability, and productivity of mining operations. Develop a robust crystal classification system that utilizes advanced technologies, such as machine learning, to accurately identify and classify mineral ores. The system should be capable of handling diverse ore compositions and geological variations.

1.5.Technology Used

Within the field of computer science, Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are related disciplines that concentrate on creating and applying models and algorithms that allow machines to carry out tasks that traditionally require human intelligence. But their approaches and scope are different. Below is a summary of each:

Intelligent artificial systems (AI):

The larger area of computer science known as artificial intelligence (AI) seeks to build machines or systems that are able to carry out operations that would normally require human intelligence. Problem-solving, reasoning, speech recognition, natural language comprehension, and decision-making are some of these tasks.

Robotics, computer vision, natural language processing, rule-based systems, expert

systems, and knowledge representation are just a few of the many subfields that fall under the umbrella of artificial intelligence (AI).

Machine Intelligence (ML):

The creation of algorithms and models that allow computers to learn from and make predictions or decisions based on data is the focus of machine learning (ML), a branch of artificial intelligence. Machine learning systems derive information and patterns from data, rather than being explicitly programmed.

ML encompasses a number of methods, including reinforcement learning, unsupervised learning, and supervised learning. By using algorithms to modify model parameters in response to data input, these techniques enhance system performance.

Email filtering, fraud detection, medical diagnosis, and recommendation engines (Netflix, Amazon) are just a few of the many uses for machine learning

Deep Learning (DL):

Deep learning (DL) is a branch of machine learning (ML) that specializes in multi-layered neural networks. These networks, which are able to automatically learn hierarchical data representations, are inspired by the composition and operations of the human brain.

Artificial neural networks (deep networks) with numerous hidden layers are the main tool used in deep learning. Transformers are utilized for tasks involving natural language processing, Recurrent Neural Networks (RNNs) for sequences, and Convolutional Neural Networks (CNNs) for image analysis.

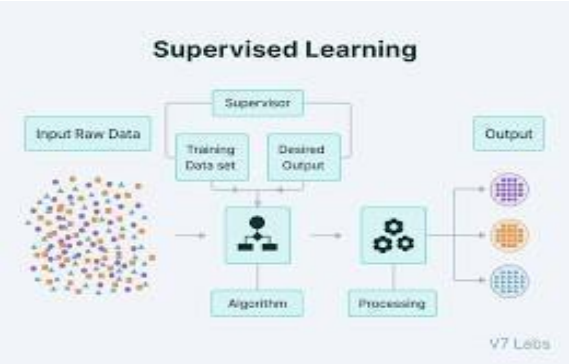


Figure 1.5.1 Supervised Learing

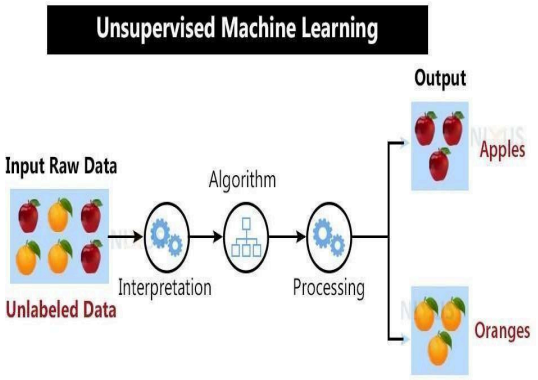


Figure 1.5.2 Unsupervised Learing

Random Forest is a powerful ensemble learning algorithm used in machine learning for both classification and regression tasks. It operates by constructing multiple decision trees during the training phase and outputting the mode (for classification) or mean prediction (for regression) of the individual trees. Each decision tree in the forest is trained on a random subset of the training data and a random subset of features, which helps to reduce overfitting and increase the model's generalization ability. During prediction, the input data pass through each decision tree, and the final prediction is made by aggregating the predictions of all trees. Random Forest is known for its high accuracy, robustness to outliers, and capability to handle large datasets with high dimensionality. It's widely used in various domains such as finance, healthcare, and bioinformatics due to its versatility and effectiveness in solving a wide range of predictive tasks.
Sure, let's delve deeper into Random Forest.

Random Forest is a versatile and powerful algorithm that belongs to the family of ensemble learning methods. It's called "ensemble" because it combines the predictions from multiple individual models, in this case, decision trees, to make a final prediction. The term "forest" refers to the collection of these decision trees.

Each decision tree in a Random Forest is constructed independently during the training phase. However, they are not identical because they are trained on different subsets of the training data and a random subset of features. This process is known as bootstrap aggregating or "bagging." By training on different subsets of data, Random Forest can capture diverse

patterns and reduce the risk of overfitting, which occurs when a model learns to memorize the training data rather than generalize from it.

Moreover, each decision tree in the Random Forest considers only a random subset of features at each split point. This further adds randomness and diversity to the trees. The rationale behind this is to decorrelate the trees from each other, making them more independent and reducing the risk of making the same mistakes.

During the prediction phase, the input data traverse through each decision tree in the forest. For classification tasks, each tree casts a "vote" for the class label, and the final prediction is determined by the majority class among all the trees. For regression tasks, each tree predicts a numerical value, and the final prediction is often the average (or mean) of all the individual tree predictions.

1.6. Software Requirement

Functional Requirements:

Graphical User Interface with the User.

Software Requirements

For developing the application the following are the Software Requirements:

Python

Django

MySql

MySqlclient

WampServer 2.4

Operating Systems supported:

Windows 7

Windows XP

Windows 8

Technologies and Languages Used to Develop

Python

Debugger and Emulator

Any Browser (Particularly Chrome)

Hardware Requirements:

For developing the application the following are the Hardware Requirements:

Processor: Pentium IV or higher

RAM: 256 MB

Space on Hard Disk: minimum 512MB

CHAPTER 2

LITERATURE REVIEW

This section covers the earlier studies that were carried out. This section contains information on machine learning algorithm developments, resilient neural network architectures, tool combinations to improve the performance of these algorithms and networks, and an overview of the creation and formulation of the neural network architectures utilized in our research. This section provides a review of relevant studies and research in this domain, highlighting key contributions and insights.

2.1. Graph Convolutional Networks for Hyperspectral Image Classification

[1]This study conducted by Maydagá and Zattin focuses on the Altar region in the Central Andes of Argentina. The researchers utilized Apatite(U–Th)/He thermochronology and Re–Os ages to understand the geological history of the area. Apatite(U–Th)/He thermochronology is a method used to determine the thermal history of rocks, while Re–Os dating is used to determine the age of mineralization events.Their findings suggest that rapid exhumation processes have occurred in the region, influencing the distribution of porphyry Cu (copper) and Au (gold) deposits. Exhumation refers to the process by which rocks are brought to the Earth's surface from deeper levels. The rapid exhumation identified in this study has implications for mineral exploration, as it may affect the concentration and accessibility of valuable metals like copper and gold.Additionally, the study provides insights into regional tectonic processes, which are the geological forces responsible for shaping the Earth's crust. Understanding these tectonic processes is crucial for interpreting the geological history of an area and predicting the distribution of mineral deposits.

[2] Hong and Gao proposed the use of Graph Convolutional Networks (GCNs) for hyperspectral image classification. Hyperspectral imaging is a technique used to collect and analyze information from across the electromagnetic spectrum, enabling detailed characterization of materials based on their spectral signatures.GCNs are a type of neural network designed to operate on graph-structured data, making them well-suited for tasks involving spatial relationships, such as image classification. By leveraging GCNs, the researchers aimed to improve the accuracy of classifying hyperspectral images, particularly in

applications related to remote sensing and mineral exploration.The significance of this work lies in its potential to enhance the interpretation of hyperspectral imagery for identifying geological features indicative of mineral deposits. Accurate classification of hyperspectral data can aid geologists and mining companies in targeting areas with high mineral potential for further exploration.

[3] Fu conducted a study on mineral prospectivity mapping of porphyry copper deposits in the Duolong Ore District of Tibet. Porphyry copper deposits are large-scale ore bodies formed by the intrusion of magma and associated hydrothermal fluids into the Earth's crust, often containing valuable metals such as copper, gold, and molybdenum.The study utilized remote sensing imagery and geochemical data to identify areas with high potential for porphyry copper mineralization. Mineral prospectivity mapping involves analyzing various geological datasets to assess the likelihood of mineral deposits occurring in specific regions.By integrating remote sensing data, which provides valuable information about surface geology and alteration patterns, with geochemical data, which offers insights into subsurface mineralization processes, the researchers aimed to create detailed maps highlighting prospective areas for porphyry copper exploration.This research is significant for mineral exploration efforts in the Duolong Ore District and beyond, as it provides a systematic approach for targeting areas with elevated potential for economically viable mineral deposits.

[4]Navarro Lorente and Miller proposed 3DeepM, an ad hoc architecture based on deep learning methods for multispectral image classification. Multispectral imagery captures data across multiple wavelengths of the electromagnetic spectrum, enabling the detection of subtle geological features that may indicate the presence of mineral deposits.Deep learning methods, particularly convolutional neural networks (CNNs), have shown promise in analyzing multispectral imagery for various applications, including mineral exploration. 3DeepM represents an advancement in this field by introducing a specialized architecture tailored to the challenges of multispectral image classification.The architecture leverages deep learning techniques to automatically learn features from multispectral data, enabling accurate classification of geological materials and terrain types. By enhancing the efficiency and accuracy of multispectral image analysis, 3DeepM has the potential to aid geologists and mining companies in identifying prospective exploration targets.This research contributes to

the ongoing efforts to leverage artificial intelligence and machine learning in mineral exploration, providing a valuable tool for analyzing multispectral imagery and identifying areas of geological interest.

2.2. More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification

[5]Hong and Gao explored the benefits of multimodal deep learning for remote sensing imagery classification. Remote sensing imagery provides valuable information about the Earth's surface, including land cover, vegetation, and geological features, which can be crucial for mineral exploration and environmental monitoring.Multimodal deep learning involves integrating data from multiple sources or modalities, such as optical imagery, radar data, and elevation maps, to improve classification accuracy and robustness. By combining complementary information from different modalities, multimodal deep learning models can effectively capture the complex relationships present in remote sensing data.The researchers demonstrated the advantages of multimodal deep learning for remote sensing imagery classification, highlighting its ability to handle diverse data sources and achieve superior performance compared to single-modal approaches. This research contributes to the development of advanced techniques for analyzing remote sensing data, with applications in mineral exploration, land use planning, and environmental management.

[6]Tapster and Costa investigated the role of crystal mush dykes as conduits for mineralizing fluids in the Yerington porphyry copper district in Nevada. Crystal mush dykes are geological features formed by the intrusion of magma into existing rock formations, resulting in a mixture of partially molten rock and solid crystals.In this study, the researchers focused on understanding how these crystal mush dykes may have served as pathways for the transportation of mineralizing fluids, which are responsible for depositing valuable minerals such as copper. By analyzing the geological characteristics of the Yerington district and conducting field observations, the researchers aimed to elucidate the mechanisms by which mineral deposits are formed in porphyry copper systems.Their findings contribute to our understanding of the geological processes governing the formation of porphyry copper deposits, particularly the role of magmatic intrusions and fluid migration pathways. Understanding these processes is essential for mineral exploration efforts, as it helps geologists identify prospective areas for further investigation.

[7]Richards and Mumin discussed magmatic-hydrothermal processes within an evolving Earth, with a focus on iron oxide-copper-gold (IOCG) and porphyry Cu ± Mo±Au deposits. Magmatic-hydrothermal processes refer to the interaction between magmatic fluids (derived from molten magma) and water-rich fluids in the Earth's crust, leading to the formation of mineral deposits.IOCG deposits and porphyry copper deposits are two types of ore systems associated with magmatic-hydrothermal processes. IOCG deposits typically contain iron oxide minerals along with copper, gold, and other metals, while porphyry copper deposits are characterized by disseminated copper mineralization associated with intrusions of igneous rock.The paper provides insights into the geological conditions and tectonic settings conducive to the formation of these types of ore deposits. By understanding the processes governing their formation, geologists can better interpret the geological history of a region and target areas with elevated potential for mineralization.

2.3. Tectonism and exhumation in convergent margin orogens: Insights from ore deposits

[8]Wilkinson and Kesler explored the relationship between tectonism (the deformation of the Earth's crust) and exhumation (the process of bringing rocks to the Earth's surface) in convergent margin orogens (mountain-building regions where tectonic plates converge). The researchers examined insights gained from studying ore deposits in these tectonically active regions.Ore deposits are often associated with tectonic processes such as mountain building, volcanic activity, and crustal deformation. Understanding the tectonic history of a region is crucial for interpreting the distribution and characteristics of ore deposits, as tectonic forces can influence the formation, preservation, and exposure of mineralized rocks.

By analyzing the geological characteristics of ore deposits in convergent margin orogens, the researchers aimed to gain insights into the tectonic processes responsible for their formation. This research contributes to our understanding of the relationship between tectonism and mineralization, with implications for mineral exploration and resource assessment in mountainous regions.

2.4. Porphyry copper systems

[9]Sillitoe provided a comprehensive overview of porphyry copper systems, which are large-scale ore deposits formed by the intrusion of magmatic fluids into the Earth's crust. Porphyry copper deposits are economically significant sources of copper, gold, and other metals, contributing to global mineral production. The paper discusses the characteristics, distribution, and tectonic controls of porphyry copper systems, drawing on decades of research and exploration experience. Porphyry copper deposits are typically associated with convergent plate boundaries, where tectonic forces lead to the emplacement of magma and the formation of mineralized hydrothermal systems. Sillitoe's work synthesizes existing knowledge about porphyry copper deposits, providing valuable insights for geologists, mining companies, and policymakers involved in mineral exploration and resource management. By understanding the geological processes governing the formation of these deposits, stakeholders can make informed decisions about exploration targeting and investment strategies.

2.5. Giant porphyry deposits: Characteristics, distribution, and tectonic controls

[10]Cooke, Hollings, and Walshe examined giant porphyry deposits, which are exceptionally large and economically significant ore bodies containing copper, gold, and other metals. These deposits represent a major source of mineral wealth globally and are often associated with convergent plate boundaries and volcanic arcs. The paper discusses the characteristics, distribution, and tectonic controls of giant porphyry deposits, drawing on case studies from around the world. The researchers examine the geological factors contributing to the formation of these deposits, including magma emplacement, hydrothermal fluid circulation, and structural controls. Understanding the characteristics and distribution of giant porphyry deposits is crucial for mineral exploration and resource assessment efforts. By identifying the geological settings conducive to their formation, geologists can target prospective areas for exploration and development, contributing to sustainable mineral resource utilization.

2.6. Triggers for the formation of porphyry ore deposits in magmatic arcs

[11]Wilkinson investigated the triggers for the formation of porphyry ore deposits in magmatic arcs, which are regions of intense volcanic and tectonic activity associated with subduction zones. Porphyry ore deposits are commonly found in magmatic arcs, where the interaction between magma and hydrothermal fluids leads to the deposition of valuable metals.The paper discusses the geological processes and factors that control the formation of porphyry ore deposits, including magma composition, fluid-rock interactions, and tectonic settings. Wilkinson examines the role of various triggers, such as changes in magma composition, crustal thickening, and faulting, in initiating mineralization events.

Understanding the triggers for porphyry ore deposit formation is essential for mineral exploration efforts, as it helps geologists identify prospective areas and prioritize exploration targets. By analyzing the geological conditions conducive to mineralization, stakeholders can make informed decisions about resource development and investment.

[12]Ahmed and Gharib investigated porphyry copper mineralization in the Eastern Desert of Egypt, utilizing geochemical analysis, alteration zoning studies, and ore mineralogy characterization. The Eastern Desert is known to host significant mineral resources, including porphyry copper deposits, which are associated with magmatic-hydrothermal systems.The researchers aimed to infer the geological processes and conditions responsible for porphyry copper mineralization in the region. By analyzing the geochemical signatures of rocks, studying alteration zones (areas where rocks have been chemically altered by hydrothermal fluids), and examining the mineralogical composition of ore deposits, they sought to unravel the history of mineralization events.Their findings contribute to our understanding of the geological factors controlling porphyry copper mineralization in the Eastern Desert of Egypt. By elucidating the processes governing ore formation, the researchers provide valuable insights for mineral exploration and resource assessment in the region, facilitating sustainable resource development.These detailed explanations offer insights into the research conducted in each referenced paper, highlighting the significance of each study for the field of mineral exploration, geological research, and resource assessment.

CHAPTER 3

METHODOLOGY

The methodology for developing a smart mining system with crystal classification of ores and industrial management involves several key steps and components. Below is a detailed outline of the methodology:

3.1.Data collection

Collecting the data requires you to collect the mineral crystals that are in the inventory. Management is responsible for administering the survey in both the current dataset and the datasets that came before it. In the input dataset, we make use of the survey that was previously collected.The machine learning model makes use of the datasets that relate to the information about ores that are obtained from Kaggle.

Geological Data: This involves gathering geological information relevant to the mining site, including geological maps, mineralogical studies, and assays. Geological maps provide insights into the geological structure, lithology, and mineralization patterns of the area. Mineralogical studies involve analyzing rock and mineral samples to identify their composition and distribution. Assays involve laboratory analysis of ore samples to determine their mineral content and grade.

Sensor Data: Sensors deployed throughout the mining site collect real-time data on various parameters such as equipment performance, environmental conditions, and geological characteristics. This includes data from equipment sensors (e.g., temperature, pressure, vibration sensors), environmental sensors (e.g., air quality, water quality sensors), and geological sensors (e.g., rock composition sensors, ore grade sensors).

3.2.Data Preprocessing

Our objective is to collect a dataset that includes information about ores, with the mining set being classified as either positive or negative. Next, it makes the necessary adjustments to the project's input dataset. I would like to inform you about the preprocessing of the dataset. First, we make sure that the values are not null, and then we create a data frame for the ML model.

Cleaning and Filtering: Raw sensor data may contain noise, outliers, or missing values that need to be addressed. Data cleaning involves removing or correcting erroneous data points, while data filtering techniques such as median filtering or Kalman filtering can help smooth out sensor noise.

Normalization and Scaling: Data normalization ensures that features are on a similar scale, preventing certain features from dominating the analysis due to differences in magnitude. Common normalization techniques include min-max scaling and z-score normalization.

Data Fusion: Data fusion involves combining data from multiple sources (e.g., geological data, remote sensing data, sensor data) to create a comprehensive dataset for analysis. Fusion techniques may include simple concatenation, weighted averaging, or more sophisticated fusion algorithms

3.3.Feature Selection

Feature extraction involves collecting a dataset and its characteristics. we are selecting the size and shape and important column of the dataset

We gather the input dataset and split it into the test and train datasets for modelling. The dataset can then be trained after preprocessing. The trained dataset can be used to make the management system. We can then accurately Predict the quantity of crystal ores in the future.

Spectral Features: In the case of hyperspectral imagery, spectral features represent the reflectance or absorption characteristics of different minerals across the electromagnetic spectrum. Techniques such as principal component analysis (PCA) or spectral angle mapping (SAM) can be used to extract relevant spectral features.

Geological Features: Geological features may include structural characteristics derived from LiDAR scans, such as fault lines, folds, or fractures. These features can provide valuable insights into the geological setting and potential mineralization zones.

Texture Features: Texture features describe spatial patterns or arrangements within the data, which can be indicative of certain geological formations or mineral distributions. Texture analysis techniques such as grey-level co-occurrence matrices (GLCM) or Gabor filters can be used to extract texture features from remote sensing imagery.

3.4.Selecting the right classifier algorithm

In an ores prediction project, Random Forest is often more suitable than Convolutional Neural Networks (CNNs) due to several key factors. Firstly, geological data relevant to ores prediction typically consists of structured features such as mineral composition, geological formations, and geochemical attributes. Random Forest's ability to handle structured/tabular data makes it well-suited for capturing complex relationships and patterns inherent in such datasets. Unlike CNNs, which are primarily designed for image-related tasks, Random Forest doesn't require data to be formatted as images, thereby eliminating the need for additional preprocessing steps to convert geological data into a suitable format for CNNs.

Furthermore, Random Forest offers several advantages in terms of interpretability and model understanding. Each decision tree in the Random Forest ensemble provides insights into the importance of different features in predicting ore presence, allowing geologists and domain experts to interpret the model's predictions and gain valuable insights into the underlying geological processes. In contrast, CNNs often operate as "black box" models, making it challenging to understand how and why specific predictions are made, especially in complex geological contexts where interpretability is crucial.

Additionally, Random Forest models are robust and perform well even with relatively small to moderate-sized datasets, which are common in ores prediction projects. Training CNNs typically requires large amounts of labeled image data, which may not always be readily available in geological studies. Moreover, Random Forest is less sensitive to outliers and noise in the data, making it more resilient to potential irregularities or uncertainties in geological datasets.

Overall, Random Forest presents a pragmatic and effective solution for ores prediction projects, offering interpretability, robust performance, and suitability for structured data analysis without the need for extensive preprocessing or large amounts of labelled data, which are common challenges in applying CNNs to geological datasets.

3.5 Prediction:

First we start our system then we start data collection from the already present dataset after that we start data preprocessing to check is their any non null value available or not if not we will proceed further if it is present then we again do the data preprocessing to remove these data for better accuracy after that we will extract the features from the dataset and divide it into training dataset which is 80% of whole dataset and 20% of whole dataset for testing so that we get to know how accurate our program is working.then we input the values of ores and mineral we want to know about once given random forest classifier starts working and looks into the trained dataset and using its algorithm to predict the ores present

Time-series Analysis: Time-series analysis techniques are used to analyze temporal patterns and trends in the dataset. This includes identifying seasonal variations, long-term trends, and potential anomalies or outliers.

Forecasting Methods: Various forecasting methods can be employed depending on the nature of the dataset and the forecasting horizon. These may include classical statistical methods such as ARIMA or exponential smoothing, machine learning algorithms such as recurrent neural networks (RNNs) or long short-term memory (LSTM) networks, or advanced deep learning models such as transformers.

Evaluation and Validation: Forecasting models need to be evaluated and validated using appropriate metrics such as mean absolute error (MAE), mean squared error (MSE), or root mean squared error (RMSE). Cross-validation techniques such as time-series cross-validation or rolling-window validation can be used to assess model performance and generalization ability.

By meticulously implementing each of these modules with attention to detail, the smart mining system can effectively leverage data collection, preprocessing, feature extraction, CNN model selection, and dataset forecasting to optimize mineral classification, mining operations, and industrial management in a holistic manner.

3.6.Matching

K-nearest neighbours (K-NN): K-NN can be used to match the features extracted from the detected face with the features stored in the database.A straightforward but powerful machine learning algorithm for classification and regression problems is K-Nearest Neighbors (K-NN). The fundamental tenet of it is that similar data points will typically lie close to one another in the feature space. The number of nearest neighbors taken into account when forming a prediction is represented by the "K" in K-NN. As a non-parametric, instance-based learning technique, K-NN assigns a new data point to the majority class among its K-nearest neighbors in the classification process. To predict a continuous output in regression, the average or weighted average of the K-nearest neighbors' target values is computed.

The simplicity and ease of implementation of K-NN is one of its main benefits. It is appropriate for online or streaming data because it does not require a training phase. Nonetheless, the ideal value of K and the selection of the distance metric—typically the Euclidean distance—have a significant impact on how well it works. Choosing the appropriate K involves making trade- offs: a large K can produce regression curves or over-smoothed boundaries, while a small K might produce noisy predictions. Support Vector Machines (SVM): SVM classifiers can also be used for matching facial features with known identities.

3.7.Database Management

Database Design:

Schema Design: Define the database schema that structures the data storage, including tables, fields, and relationships. This schema should accommodate various types of data, such as geological data, sensor data, remote sensing data, and industrial management data.

Normalization: Apply normalization techniques to eliminate data redundancy and ensure data integrity. Normalize the database to reduce the risk of anomalies and inconsistencies in the data.

Indexing: Create indexes on key fields to improve data retrieval performance, especially for frequently accessed data. Proper indexing can speed up query execution and enhance overall database performance.

Data Storage and Retrieval:

Data Storage Architecture: Select an appropriate data storage architecture based on the requirements of the project. This may include relational databases (e.g., MySQL, PostgreSQL), NoSQL databases (e.g., MongoDB, Cassandra), or a combination of both for handling structured and unstructured data.

Data Partitioning: Implement data partitioning strategies to distribute data across multiple storage devices or servers for scalability and performance optimization. Partitioning can be based on factors such as geographical location, time intervals, or data access patterns.

Data Compression and Encryption: Apply data compression techniques to reduce storage space requirements and optimize data transfer over the network. Implement data encryption mechanisms to ensure data security and confidentiality, especially for sensitive information.

Data Integration and ETL:

Data Integration: Integrate data from heterogeneous sources, including geological surveys, remote sensing systems, sensor networks, and industrial management systems. Develop data integration pipelines to consolidate data from disparate sources into a unified database.

ETL (Extract, Transform, Load): Design and implement ETL processes to extract data from the source systems, transform it into a suitable format, and load it into the database. Apply data cleansing, transformation, and enrichment techniques during the ETL process to ensure data quality and consistency.

Data Access and Security

Access Control: Implement access control mechanisms to regulate user access to the database and ensure data security. Define roles and permissions for different user groups based on their roles and responsibilities within the organization.

Authentication and Authorization: Authenticate users and authorize their access to specific data based on their credentials and privileges. Implement robust authentication mechanisms such as multi-factor authentication (MFA) to prevent unauthorized access to the database.

Auditing and Logging: Enable auditing and logging functionalities to track data access and modifications. Maintain audit trails to monitor user activities and detect any unauthorized or suspicious behavior.

Backup and Recovery:

Backup Strategy: Develop a backup strategy to regularly back up the database to prevent data loss in case of hardware failures, software errors, or other disasters. Implement full backups, differential backups, and transaction log backups to ensure comprehensive data protection.

Disaster Recovery: Establish disaster recovery procedures to restore the database to a consistent state in the event of a catastrophic failure. Maintain off-site backups and implement failover mechanisms to minimize downtime and ensure business continuity.

Performance Optimization:

Query Optimization: Optimize database queries to improve query performance and reduce response times. Analyze query execution plans, identify performance bottlenecks, and apply optimization techniques such as index optimization, query rewriting, and query caching.

Database Tuning: Fine-tune database parameters and configuration settings to optimize resource utilization and enhance database performance. Monitor system metrics such as CPU usage, memory usage, and disk I/O to identify and address performance issues proactively.

Scalability and Replication:

Horizontal Scaling: Implement horizontal scaling strategies to distribute database workload across multiple servers or nodes. Use techniques such as sharding or partitioning to scale out the database infrastructure and accommodate growing data volumes.

Replication: Set up database replication to create redundant copies of the database for fault tolerance and high availability. Configure master-slave or master-master replication topologies to replicate data across geographically dispersed locations.

Monitoring and Maintenance:

Database Monitoring: Deploy database monitoring tools to monitor database performance, track resource usage, and identify potential issues. Set up alerts and notifications to proactively address performance degradation or system failures.

Routine Maintenance: Schedule routine maintenance tasks such as database backups, index rebuilds, and statistics updates to keep the database healthy and optimized. Perform regular database health checks and performance audits to ensure optimal database performance.

By implementing a robust Database Management module as part of the smart mining system, mining companies can effectively organize and manage the vast amount of data generated from geological surveys, sensor networks, remote sensing systems, and industrial management processes. This enables efficient data storage, retrieval, integration, security, and performance optimization, ultimately facilitating informed decision-making and resource optimization in mining operations.

3.8.Real-Time Processing

The real-time processing module begins with the continuous acquisition of data from various sources across the mining operation. This includes sensor data from mining equipment, environmental sensors, geological sensors, as well as data streams from remote sensing devices such as drones or satellite imagery.As data is collected in real-time, it is streamed to a centralized data processing system. This system is responsible for receiving, processing, and analyzing the incoming data streams connuously.Upon receiving the data streams, preprocessing steps are applied in real-time to clean, filter, and transform the raw data into a format suitable for analysis. This involves handling missing values, removing outliers, and performing necessary data transformations on-the-fly.

Real-time feature extraction techniques may be applied to extract relevant features from the incoming data streams. Feature selection methods are then employed to identify the most informative features for downstream analysis and decision-making processes.Utilizing machine learning models or statistical algorithms capable of real-time analysis, predictive analytics are applied to forecast future trends, detect anomalies, or predict equipment failures. These predictive models continuously update and adapt based on the latest data available.

In the context of crystal classification of ores, real-time processing enables the classification and identification of minerals based on their spectral signatures or other features extracted from hyperspectral imagery or sensor data. Advanced machine learning techniques, such as convolutional neural networks (CNNs), may be deployed for real-time classification tasks.

Real-time monitoring dashboards provide stakeholders with up-to-date insights into key performance indicators (KPIs), equipment status, environmental conditions, and ore quality. Alerts and notifications are triggered in real-time to flag any deviations from expected norms or potential safety hazards.

Control mechanisms integrated into the system enable real-time adjustments to mining operations based on the insights gained from data analysis. This may include optimizing drilling patterns, adjusting equipment settings, or redirecting resources to areas with higher mineral potential.

The real-time processing module seamlessly integrates with industrial management systems, such as Enterprise Resource Planning (ERP) or Mine Planning and Management (MPM) software. This ensures that real-time insights derived from data analysis are translated into actionable decisions and integrated into broader operational workflows.

The real-time processing module is designed to be scalable and capable of handling large volumes of data streams efficiently. This involves deploying distributed computing architectures or cloud-based solutions to ensure optimal performance and responsiveness even as data volumes increase.

The real-time processing module undergoes continuous optimization and refinement based on feedback from stakeholders and ongoing performance evaluations. This includes fine-tuning algorithms, improving data processing pipelines, and enhancing system scalability and reliability over time.

By incorporating real-time processing capabilities into the smart mining system, mining companies can gain actionable insights, optimize resource utilization, improve safety, and enhance overall operational efficiency in a dynamic and rapidly changing environment.

In the context of the "Smart mining system with crystal classification of ores and industrial management" project, post-processing and decision-making play crucial roles in converting raw data into actionable insights and guiding strategic and operational decisions. Here's an explanation of post-processing and decision-making within this project:

3.9.Post-Processing and Decision-Making

After real-time processing, data from various sources such as sensor data, geological surveys, remote sensing imagery, and equipment telemetry are aggregated and consolidated. This ensures that all relevant data is available for analysis and decision-making.

Post-processing involves further cleaning and validation of the processed data to ensure accuracy and reliability. This may include identifying and correcting errors, handling missing values, and verifying the consistency of the data across different sources.

Feature engineering techniques are applied to extract additional features or transform existing features to enhance the predictive power of the data. This may involve deriving new features from existing ones, scaling or normalizing features, or encoding categorical variables.

Statistical analysis techniques are used to uncover patterns, trends, and relationships within the data. Descriptive statistics, correlation analysis, and hypothesis testing help to identify significant insights. Data visualization tools such as charts, graphs, and heatmaps are employed to visually represent the findings and make them more accessible to stakeholders.

Post-processing includes the application of advanced analytics techniques such as pattern recognition and clustering to identify hidden patterns or groups within the data. This can aid in identifying anomalies, detecting trends, and segmenting the data into meaningful clusters for further analysis.

If machine learning models were employed for predictive analytics during real-time processing, post-processing involves evaluating the performance of these models using validation techniques such as cross-validation or holdout validation. Model parameters may be fine-tuned or optimized to improve predictive accuracy and generalization ability.

Decision-Making:

Decision-making at a strategic level involves setting long-term goals and defining the overall direction of the mining operation. This may include decisions related to resource allocation, exploration strategies, investment priorities, and sustainability initiatives.

Operational decision-making focuses on optimizing day-to-day activities within the mining operation to maximize efficiency and productivity. This includes decisions

related to production scheduling, equipment deployment, maintenance planning, and supply chain management.

Decision-making in mineral classification involves determining the quality and quantity of mineral deposits based on the analysis of ore samples and geological data. This information informs decisions about mine planning, ore extraction methods, and resource utilization.

Decisions related to environmental management and regulatory compliance are critical for sustainable mining operations. This includes decisions regarding waste management, water conservation, air quality monitoring, and adherence to environmental regulations. Decision-making involves identifying and assessing potential risks and uncertainties associated with mining operations, such as geological hazards, equipment failures, market volatility, and regulatory changes. Strategies for risk mitigation and contingency planning are developed to minimize the impact of these risks.

Continuous monitoring of key performance indicators (KPIs) enables stakeholders to assess the effectiveness of decisions and identify areas for improvement. Feedback loops are established to incorporate lessons learned into future decision-making processes, ensuring ongoing optimization and adaptation.

By leveraging post-processing techniques and informed decision-making, the smart mining system can transform raw data into actionable insights, enabling stakeholders to make informed decisions that drive operational efficiency, maximize resource utilization, and ensure the sustainability of mining operations.

3.10.System Design

Designing a system for the "Smart mining system with crystal classification of ores and industrial management" project involves integrating various components and technologies to achieve the project objectives effectively. Here's a high-level overview of the system design:

Overall Architecture The system architecture should be modular and scalable, allowing for the integration of different modules and components. It should support both real-time processing and batch-processing workflows to accommodate different use cases and data processing requirements. Implement data collection mechanisms to gather data from

various sources such as sensors, geological surveys, remote sensing devices, and equipment telemetry.Integrate data from different sources into a centralized data management platform or data lake for storage and processing. Use standardized data formats and protocols to ensure interoperability and compatibility across different data sources and systems. Develop a real-time processing module to handle streaming data from sensors and other sources. Implement data preprocessing, feature extraction, and predictive analytics algorithms to analyze incoming data streams in real-time Utilize scalable and distributed processing frameworks such as Apache Kafka, Apache Flink, or Apache Storm to handle large volumes of streaming data efficiently.

Batch Processing Module:

Develop a batch processing module for analyzing historical data and performing offline data processing tasks.Use distributed computing frameworks such as Apache Hadoop or Apache Spark for parallel processing of large datasets.Implement batch processing workflows for tasks such as data cleaning, feature engineering, model training, and evaluation.

Machine Learning Models:

Employ machine learning models for tasks such as mineral classification, predictive maintenance, anomaly detection, and forecasting. Utilize algorithms such as convolutional neural networks (CNNs), support vector machines (SVMs), decision trees, and ensemble methods based on the specific requirements of each task. Train and deploy machine learning models using frameworks such as TensorFlow, PyTorch, or Scikit-learn.

Decision Support System

Develop a decision support system to assist stakeholders in making informed decisions based on the insights derived from data analysis. Provide interactive dashboards, visualization tools, and reporting functionalities to present data and analysis results in a user-friendly manner. Incorporate alerting mechanisms to notify stakeholders about critical events, anomalies, or deviations from expected norms.

Integration with Industrial Management Systems:

Integrate the smart mining system with existing industrial management systems such as Enterprise Resource Planning (ERP) or Mine Planning and Management (MPM) software. Develop APIs and data connectors to facilitate seamless data exchange and interoperability between the smart mining system and other systems. Enable

bi-directional communication to allow industrial management systems to trigger actions or updates based on insights from the smart mining system.

Security and Compliance:

Implement robust security measures to protect sensitive data and ensure compliance with data privacy regulations. Use encryption, access control mechanisms, and audit trails to secure data at rest and in transit. Conduct regular security assessments and audits to identify and mitigate potential vulnerabilities.

Scalability and Performance:

Design the system to be scalable and capable of handling increasing volumes of data and processing demands over time.Utilize cloud computing services such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP) to leverage on-demand scalability and resource elasticity.Optimize system performance through parallel processing, distributed computing, and efficient resource utilization.

Monitoring and Maintenance:

Implement monitoring and logging mechanisms to track system performance, resource utilization, and data quality.Set up automated alerts and notifications to proactively identify and address issues such as hardware failures, software errors, or data inconsistencies.Establish regular maintenance routines to update software components, patch security vulnerabilities, and fine-tune system parameters for optimal performance.

By following this system design, the smart mining system can effectively integrate data collection, real-time processing, machine learning, decision support, and integration with industrial management systems to enable efficient mineral classification, mining operations, and industrial management.

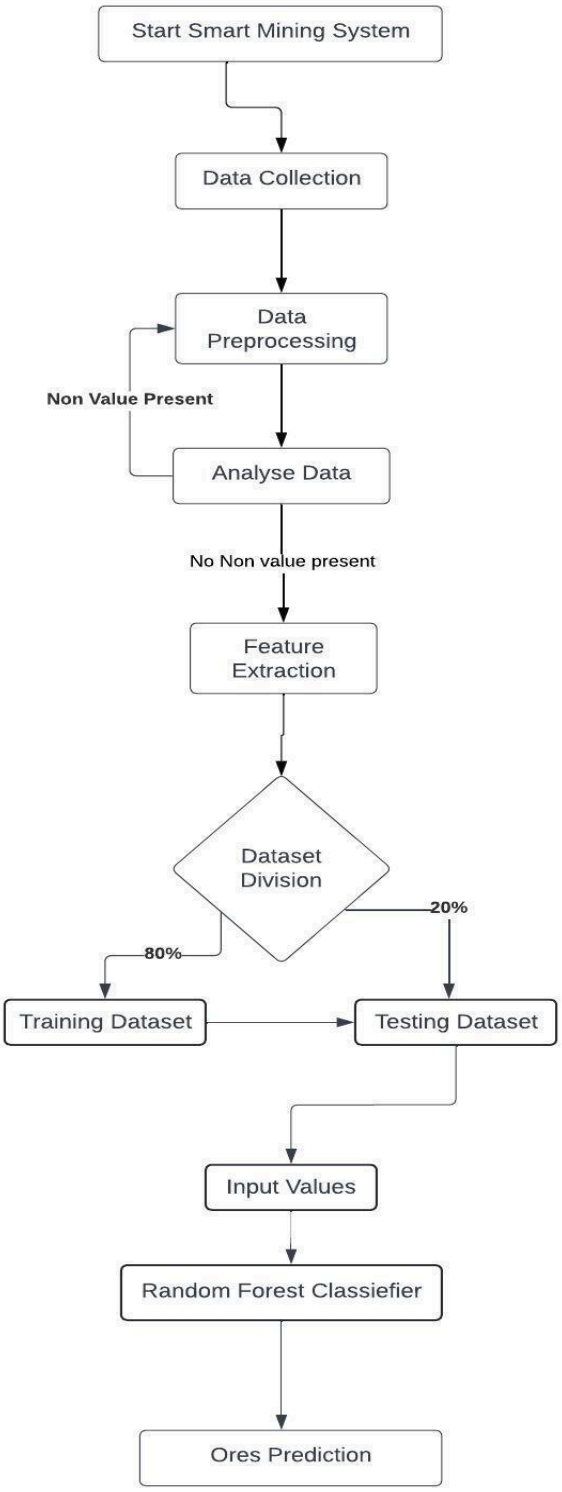


Figure 3.10.1 Architecture Diagram

an architecture diagram for a smart mining system with crystal classification of ores and industrial management involves illustrating the key components and their interactions.

Components:

Mining Site:Represent the physical mining site where ore extraction takes place. Include sensors for data collection and mining equipment.

Sensor System:

Illustrate sensors for collecting data on ore composition, quality, and quantity. Specify the types of sensors, such as spectroscopy or imaging sensors.

Ore Processing Unit:Represent the unit responsible for initial ore processing. Include machinery for crushing, sorting, and preparing ore samples.

Crystal Classification Module: Show the module responsible for classifying crystals based on the collected ore samples.Include algorithms or AI models for crystal classification.

Industrial Management Server: Illustrate the server responsible for managing and coordinating industrial processes.Include a database for storing classified crystal data and industrial management information.

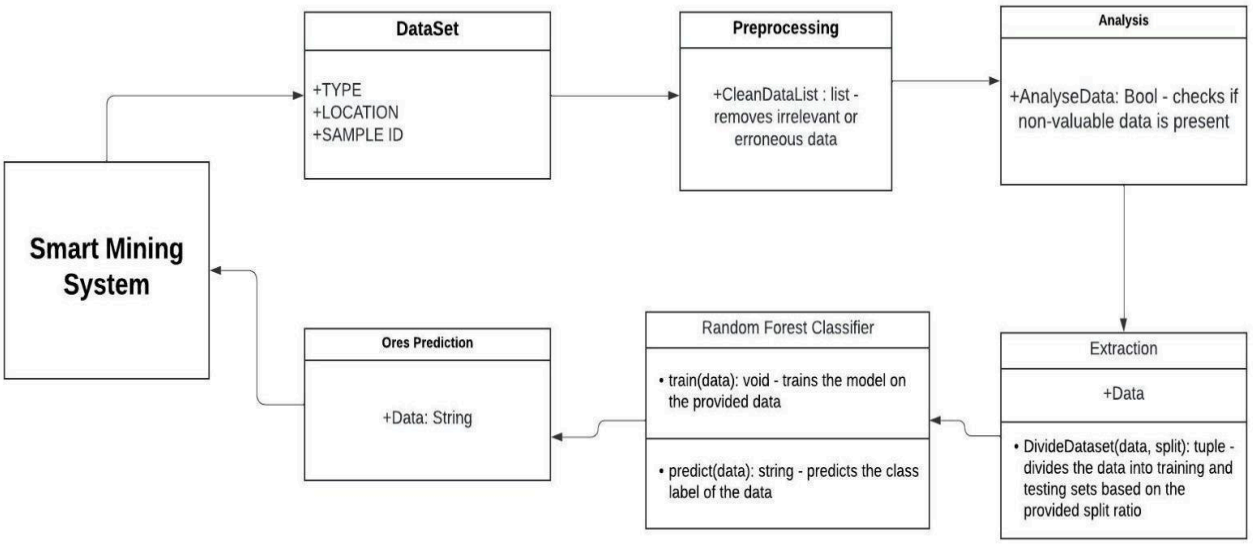


Figure.3.10.2 UML Diagram

illustrate the flow of ore from the mining site to the Ore Processing Unit.

Connect Ore Processing Unit to Crystal Classification Module:

Show the flow of processed ore samples to the Crystal Classification Module for crystal analysis.

Connect Crystal Classification Module to Industrial Management Server:

Illustrate how the classified crystal data is transferred to the Industrial Management Server.

Connect Industrial Management System to Industrial Management Server:
Show the connection between the software system and the server for managing industrial processes.

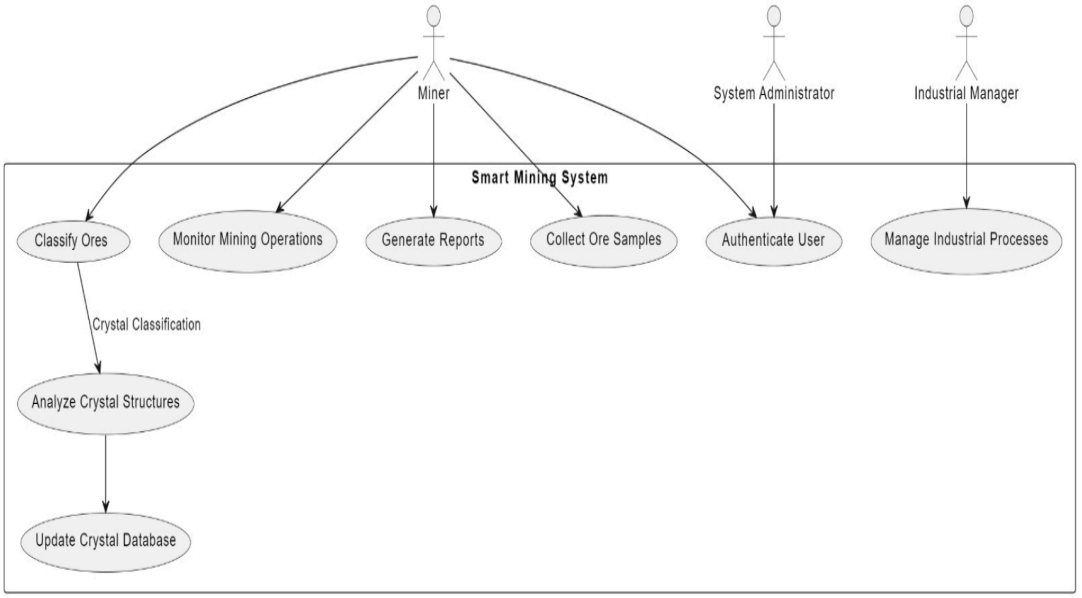


Figure 3.10.3 Usecase Diagram

3.11.Algorithms used

In machine learning, Random Forest is a potent ensemble learning algorithm that is applied to both regression and classification problems. During the training phase, it builds multiple decision trees and outputs the mean prediction (for regression) or mode (for classification) of each tree. To lessen overfitting and improve the model's capacity for generalisation, each decision tree in the forest is trained using a random subset of features and training data. Each decision tree in the prediction process processes the input data, and the ultimate prediction is produced by adding the predictions from each tree. Random Forest is renowned for its high dimensionality, high accuracy, and resilience to outliers in large datasets. Because of its adaptability and efficiency in handling a broad range of predictive tasks, it is extensively utilised in many different fields, including bioinformatics, finance, and healthcare.

One of the more potent and adaptable algorithms in the ensemble learning toolkit is Random Forest. The term "ensemble" refers to the fact that it synthesises the predictions

from several different models—in this case, decision trees—to produce a single final prediction. These decision trees are referred to as a "forest" collectively.

In the training stage of a Random Forest, every decision tree is built separately. They are not the same, though, since they are trained using distinct subsets of the training set and a randomised subset of the features. This procedure is called "bagging" or bootstrap aggregating. Random Forest is able to identify a variety of patterns and lower the possibility of overfitting, which happens when a model learns to memorise the training data instead of drawing generalisations from it, by training on various subsets of data.

Furthermore, at each split point, each Random Forest decision tree only takes into account a random subset of features. This gives the trees even more diversity and randomness. This is done with the intention of decorrelate the trees from one another, increasing their independence and decreasing the likelihood that they will make the same errors.

The input data pass through each decision tree in the forest during the prediction phase. In classification tasks, the majority class among all the trees determines the final prediction, with each tree casting a "vote" for the class label. Each tree in a regression task predicts a numerical value; the final prediction is frequently the average, or mean, of all the predictions made by each tree.

We trained the model with different algorithm to find out the algorithm with the highest accuracy

Random Forest algorithm often outperforms a single Decision Tree due to several key advantages. Firstly, while Decision Trees are prone to overfitting, Random Forest mitigates this issue by combining multiple decision trees trained on different subsets of the data. This ensemble approach helps to reduce variance and improve generalization, leading to more robust and reliable predictions

Random Forest often outperforms Gaussian Naive Bayes (GNB) in various contexts due to its ability to handle complex relationships and high-dimensional data more effectively.

Random Forest often outperforms Logistic Regression in various scenarios due to its ability to capture complex nonlinear relationships and handle high-dimensional datasets more effectively. Unlike Logistic Regression

CHAPTER 4

RESULTS AND DISCUSSION

Data collection: Collect the crystal of ores in the management. The management gives the survey in the present and past datasets. We use the survey of the collection data set in the input dataset.

We Collect ore production data from KAGGLE,then convert them in CSV File and from the collected data we train and forecast the upcoming Ores Production

Total of 737453 data set has been collected stored in the data stored in the dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	date	% Iron Fee	% Silica Fe	Starch Flo	Amina Flo	Ore Pulp F	Ore Pulp p	Ore Pulp D	Flotation C	Flotation C	Flotation C	Flotation C	Flotation C	Flotation C	Flotation C	Flotation C	Flotation C	Flotation C	Flotation C
2	########	55,2	16,98	3019,53	5,57,434	3,95,713	1,00,664	1,74	2,49,214	2,53,235	2,50,576	2,95,096	306,4	2,50,225	2,50,884	4,57,396	4,32,962	4,24,954	4,43,558
3	########	55,2	16,98	3024,41	5,63,965	3,97,383	1,00,672	1,74	2,49,719	2,50,532	2,50,862	2,95,096	306,4	2,50,137	2,48,994	4,51,891	429,56	4,32,939	4,48,086
4	########	55,2	16,98	3043,46	5,68,054	3,99,668	10,068	1,74	2,49,741	2,47,874	2,50,313	2,95,096	306,4	2,51,345	2,48,071	451,24	4,68,927	434,61	4,49,688
5	########	55,2	16,98	3047,36	5,68,665	3,97,939	1,00,689	1,74	2,49,917	2,54,487	2,50,049	2,95,096	306,4	2,50,422	2,51,147	4,52,441	4,58,165	4,42,865	446,21
6	########	55,2	16,98	3033,69	5,58,167	4,00,254	1,00,697	1,74	2,50,203	2,52,136	2,49,895	2,95,096	306,4	2,49,983	2,48,928	4,52,441	452,9	4,50,523	453,67
7	########	55,2	16,98	3079,1	5,64,697	3,96,533	1,00,705	1,74	250,73	2,48,906	2,49,521	2,95,096	306,4	2,50,356	2,51,873	4,44,384	4,43,269	4,60,449	439,92
8	########	55,2	16,98	3127,79	5,66,467	392,9	1,00,713	1,74	2,50,313	2,52,202	2,49,082	2,95,096	306,4	250,95	2,53,477	4,46,185	4,44,571	4,52,306	4,31,328
9	########	55,2	16,98	3152,93	5,58,777	3,97,002	1,00,722	1,74	2,49,895	253,63	2,49,258	2,95,096	306,4	2,49,456	2,53,345	4,45,985	4,61,341	461,64	4,42,067
10	########	55,2	16,98	3147,27	556,03	3,94,307	10,073	1,74	2,50,137	2,51,104	2,48,774	2,95,096	306,4	2,48,577	2,50,884	4,46,686	4,78,385	4,59,103	4,55,074
11	########	55,2	16,98	3142,58	5,65,857	3,93,105	1,00,738	1,74	2,49,653	2,52,202	2,48,203	2,95,096	306,4	2,48,511	2,48,137	4,45,685	4,78,779	4,60,665	4,57,225
12	########	55,2	16,98	3148,05	5,61,951	3,96,533	1,00,746	1,74	2,49,236	2,50,818	2,50,225	2,95,096	306,4	2,50,203	2,46,797	4,56,495	438,06	4,66,332	4,58,005
13	########	55,2	16,98	3150,39	5,58,472	3,97,852	1,00,755	1,74	249,17	2,49,829	2,51,147	2,95,096	306,4	2,50,928	2,46,533	461,45	421,41	467,79	458,59
14	########	55,2	16,98	3280,27	5,64,026	3,93,545	1,00,763	1,74	2,49,016	2,49,829	2,51,147	2,95,096	306,4	2,49,543	2,51,147	4,57,947	4,25,372	4,53,818	4,53,942
15	########	55,2	16,98	3411,13	5,67,261	394,16	1,00,771	1,74	2,49,258	2,50,137	2,51,609	2,95,096	306,4	2,48,643	2,49,587	4,48,037	428,26	4,47,074	4,58,516
16	########	55,2	16,98	3447,46	5,61,646	3,92,549	1,00,779	1,74	249,39	2,51,191	2,50,269	2,95,096	306,4	2,49,434	2,50,225	4,33,923	4,18,459	4,31,266	4,65,705
17	########	55,2	16,98	3562,7	5,60,364	3,94,688	1,00,788	1,74	2,50,005	2,52,202	2,49,456	2,95,096	306,4	2,50,598	248,84	4,34,674	416,42	4,35,956	4,59,922
18	########	55,2	16,98	3707,03	5,63,049	3,96,504	1,00,796	1,74	2,50,115	249,39	2,49,697	2,95,096	306,4	2,51,082	2,50,774	4,46,736	3,98,407	4,44,212	444,88
19	########	55,2	16,98	3784,96	5,57,983	3,94,834	1,00,804	1,74	2,50,049	2,46,533	2,49,829	2,95,096	306,4	2,50,378	2,48,643	4,51,991	4,26,335	4,43,024	4,29,998
20	########	55,2	16,98	3798,05	563,11	3,96,709	1,00,812	1,74	2,50,203	2,48,181	2,50,291	2,95,096	306,4	2,50,203	2,49,807	4,47,537	4,28,318	4,45,292	425,78
21	########	55,2	16,98	3866,6	564,27	3,98,262	1,00,821	1,74	2,50,269	2,47,939	2,49,719	2,95,096	306,4	2,50,422	2,51,543	4,52,441	4,43,155	4,44,104	4,18,668
22	########	55,2	16,98	3907,42	5,68,054	3,94,951	1,00,829	1,74	2,50,115	2,49,258	2,49,697	2,95,096	306,4	2,50,642	2,48,884	4,39,679	4,51,199	4,46,372	4,20,548
23	########	55,2	16,98	3705,66	5,60,669	3,96,123	1,00,837	1,74	2,50,356	2,51,433	2,49,521	2,95,096	306,4	250,4	2,49,324	4,41,681	446,1	4,44,428	4,21,288
24	########	55,2	16,98	3554,3	5,72,449	395,01	1,00,845	1,74	2,50,488	2,47,104	2,49,236	2,95,096	306,4	2,50,115	249,28	4,47,186	4,68,245	4,51,657	4,49,687
25	########	55,2	16,98	3523,24	5,63,416	3,94,453	1,00,854	1,74	2,50,181	2,47,456	2,49,324	2,95,096	306,4	2,49,434	2,48,181	4,53,542	5,00,023	4,54,624	4,58,008
26	########	55,2	16,98	3514,65	5,65,613	4,00,986	1,00,862	1,74	249,28	2,52,971	2,51,697	2,95,096	306,4	2,49,324	2,51,543	450,69	4,92,148	4,58,458	4,58,714

Figure 4.1.1 Dataset

pandas library to read the CSV file into a DataFrame, which is a tabular data structure in Python.Now we read a CSV file containing data related to a mining process flotation plant into a pandas DataFrame, enabling further analysis and manipulation of the data using Python.

	date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	...	Flotation Column 07 Air Flow	Flotation Column 01 Level	Flotation Column 02 Level	Flotation Column 03 Level	Flotation Column 04 Level	Flotation Column 05 Level	Flotation Column 06 Level	Flotation Column 07 Level
0	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	...	250.884	457.396	432.962	424.954	443.558	502.255	446.370	523.344
1	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	...	248.994	451.891	429.560	432.939	448.086	496.363	445.922	498.075
2	2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874	...	248.071	451.240	468.927	434.610	449.688	484.411	447.826	458.567
3	2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487	...	251.147	452.441	458.165	442.865	446.210	471.411	437.690	427.669
4	2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136	...	248.928	452.441	452.900	450.523	453.670	462.598	443.682	425.679

Figure 4.1.2 DataFrame

Then We check for non null value in the dataframe because is used to provide an overview of the DataFrame df, including its structure and content, as well as to quantify the total number of missing values present in the DataFrame. Understanding missing values is crucial for data preprocessing and analysis as it helps in deciding how to handle them, such as imputation or removal, before performing further analysis or modeling

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 737453 entries, 0 to 737452
Data columns (total 23 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   % Iron Feed                   737453 non-null  float64
 1   % Silica Feed                 737453 non-null  float64
 2   Starch Flow                   737453 non-null  float64
 3   Amina Flow                    737453 non-null  float64
 4   Ore Pulp Flow                 737453 non-null  float64
 5   Ore Pulp pH                   737453 non-null  float64
 6   Ore Pulp Density              737453 non-null  float64
 7   Flotation Column 01 Air Flow  737453 non-null  float64
 8   Flotation Column 02 Air Flow  737453 non-null  float64
 9   Flotation Column 03 Air Flow  737453 non-null  float64
 10  Flotation Column 04 Air Flow  737453 non-null  float64
 11  Flotation Column 05 Air Flow  737453 non-null  float64
 12  Flotation Column 06 Air Flow  737453 non-null  float64
 13  Flotation Column 07 Air Flow  737453 non-null  float64
 14  Flotation Column 01 Level     737453 non-null  float64
 15  Flotation Column 02 Level     737453 non-null  float64
 16  Flotation Column 03 Level     737453 non-null  float64
 17  Flotation Column 04 Level     737453 non-null  float64
 18  Flotation Column 05 Level     737453 non-null  float64
 19  Flotation Column 06 Level     737453 non-null  float64
 20  Flotation Column 07 Level     737453 non-null  float64
 21  % Iron Concentrate            737453 non-null  float64
 22  % Silica Concentrate          737453 non-null  float64
dtypes: float64(23)
memory usage: 129.4 MB
(None, 0)
```

Figure 4.1.3 Non-Null Value Result

Then descriptive statistics of the DataFrame df, focusing on numerical columns and readable summary of descriptive statistics for numerical columns in the DataFrame, facilitating exploratory data analysis and informing subsequent data processing steps or modeling decisions.

	count	mean	std	min	25%	50%	75%	max
% Iron Feed	737453.0	56.294739	5.157744	42.740000	52.670000	56.080000	59.720000	65.78000
% Silica Feed	737453.0	14.651716	6.807439	1.310000	8.940000	13.850000	19.600000	33.40000
Starch Flow	737453.0	2869.140569	1215.203734	0.002026	2076.320000	3018.430000	3727.730000	6300.23000
Amina Flow	737453.0	488.144697	91.230534	241.669000	431.796000	504.393000	553.257000	739.53800
Ore Pulp Flow	737453.0	397.578372	9.699785	376.249000	394.264000	399.249000	402.968000	418.64100
Ore Pulp pH	737453.0	9.767639	0.387007	8.753340	9.527360	9.798100	10.038000	10.80810
Ore Pulp Density	737453.0	1.680380	0.069249	1.519820	1.647310	1.697600	1.728330	1.85325
Flotation Column 01 Air Flow	737453.0	280.151856	29.621288	175.510000	250.281000	299.344000	300.149000	373.87100
Flotation Column 02 Air Flow	737453.0	277.159965	30.149357	175.156000	250.457000	296.223000	300.690000	375.99200
Flotation Column 03 Air Flow	737453.0	281.082397	28.558268	176.469000	250.855000	298.696000	300.382000	364.34600
Flotation Column 04 Air Flow	737453.0	299.447794	2.572538	292.195000	298.262566	299.805000	300.638000	305.87100
Flotation Column 05 Air Flow	737453.0	299.917814	3.636579	286.295000	298.068000	299.887120	301.791137	310.27000
Flotation Column 06 Air Flow	737453.0	292.071485	30.217804	189.928000	262.541000	299.477000	303.061000	370.91000
Flotation Column 07 Air Flow	737453.0	290.754856	28.670105	185.962000	256.302000	299.011000	301.904000	371.59300
Flotation Column 01 Level	737453.0	520.244823	131.014924	149.218000	416.978000	491.878000	594.114000	862.27400
Flotation Column 02 Level	737453.0	522.649555	128.165050	210.752000	441.883000	495.956000	595.464000	828.91900
Flotation Column 03 Level	737453.0	531.352662	150.842164	126.255000	411.325000	494.318000	601.249000	886.82200
Flotation Column 04 Level	737453.0	420.320973	91.794432	162.201000	356.679000	411.974000	485.549000	680.35900

Figure 4.1.4 Descriptive Statistics of the DataFrame

Then we used for data preprocessing and splitting, which is a crucial step in machine learning model development. It ensures that the data is appropriately divided into training, validation, and test sets, facilitating model training, evaluation, and tuning.This splitting is essential for assessing the performance of machine learning models on unseen data and preventing overfitting. Similar to the previous splitting, it randomly shuffles the data and assigns 80% to the training split and 20% to the validation split. This validation split is crucial for tuning hyperparameters and monitoring the model's performance during training to prevent overfitting.

```
[15]
    df= df.drop(df.columns[[0]], axis=1)
    train, test = train_test_split(df,test_size=0.2,random_state=42)
    train_split,val_split=train_test_split(train,test_size=0.2,random_state=42)
    train.shape,test.shape,train_split.shape,val_split.shape

...  ((589962, 23), (147491, 23), (471969, 23), (117993, 23))
```

Figure 4.1.5 Training and Testing Set

The heatmap is a visual representation of the correlation matrix of the train DataFrame. It shows the correlation coefficients between each pair of features in the dataset.

Here's a breakdown of the different elements of the heatmap:

Color: The color of each cell represents the correlation coefficient between the two features corresponding to that cell. The color scale ranges from blue (negative correlation) to white (no correlation) to red (positive correlation). The specific colors used are from a diverging palette, where the colors closer to the center (white) represent weaker correlations, and the colors at the extremes (blue and red) represent stronger correlations.

Values: The numerical values of the correlation coefficients are displayed within each cell. These values are formatted to two decimal places using the fmt='.2f' argument.

Annotations: The heatmap also includes annotations, which are the actual correlation coefficients displayed within each cell. These annotations are enabled using the annot=True argument and their font size is set to 8 using the annot_kws={'size': 8} argument.

Interpreting the heatmap:

Positive correlation: A positive correlation coefficient (closer to 1) indicates that the two features tend to move in the same direction. For example, if two features have a correlation coefficient of 0.8, it means that when one feature increases, the other feature also tends to increase.

Negative correlation: A negative correlation coefficient (closer to -1) indicates that the two features tend to move in opposite directions. For example, if two features have a correlation coefficient of -0.7, it means that when one feature increases, the other feature tends to decrease.

No correlation: A correlation coefficient close to 0 indicates that there is no linear relationship between the two features.

This heatmap can be a useful tool for exploring the relationships between features in your data. By looking for patterns in the colors and values, you can identify which features are highly correlated or not correlated at all. This information can be helpful for tasks such as feature selection, model building, and understanding the underlying structure of your data.

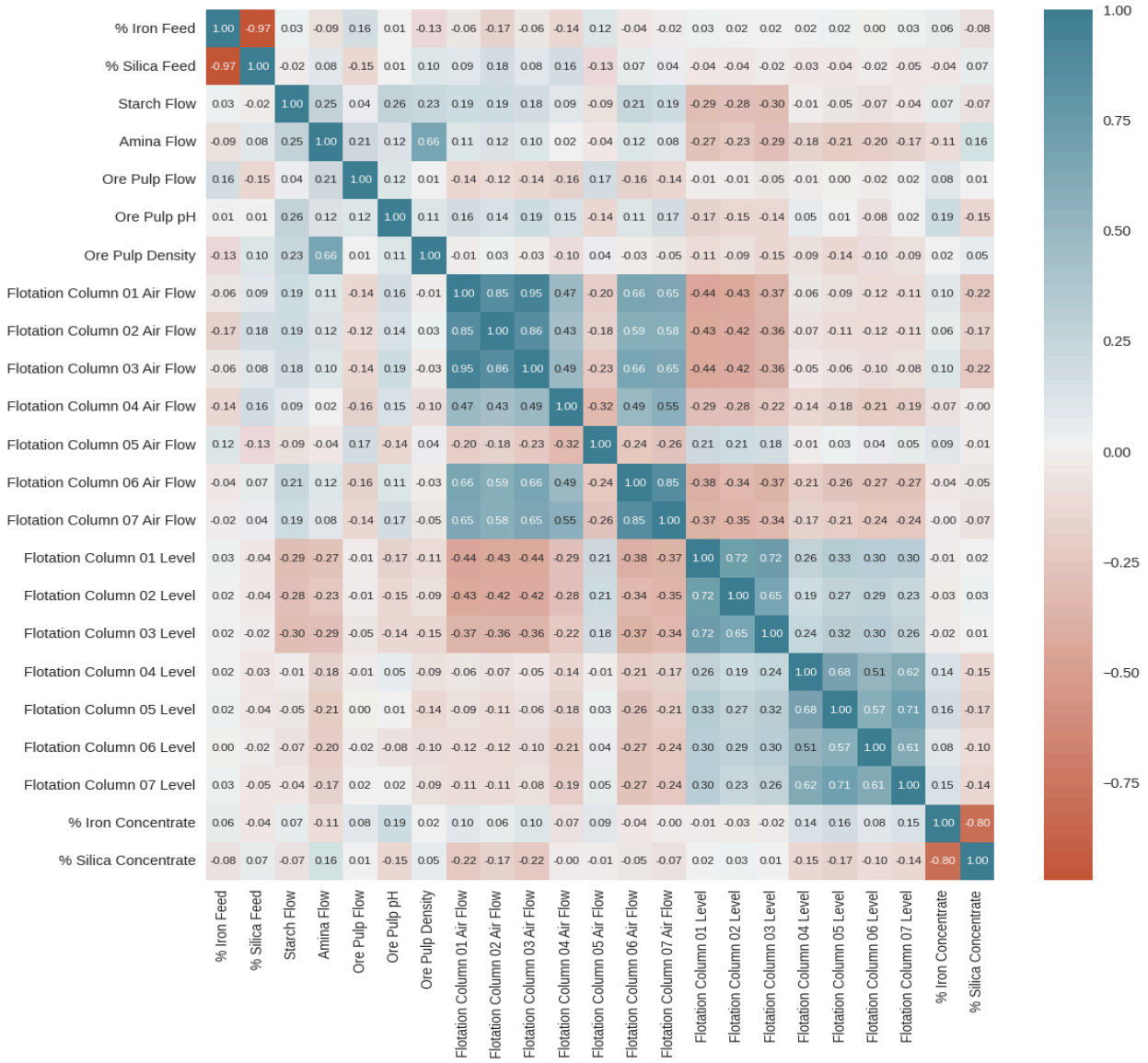


Figure 4.1.6 Heatmap Visual Representation Of The Correlation Matrix Of The Train DataFrame.

The function three subplots within a single figure and displays them in a grid layout. Each subplot visualizes a different aspect of the distribution of the specified feature:

1. **Histogram:** The first subplot shows a histogram of the feature's values. The `norm_hist=True` argument ensures that the histogram is normalized to represent a probability density function.
2. **QQ-plot:** The second subplot shows a quantile-quantile (QQ) plot of the feature's values. This plot compares the distribution of the feature to a theoretical normal distribution. If the points in the QQ-plot fall close to a straight line, it suggests that the feature's distribution is close to normal.
3. **Box plot:** The third subplot shows a box plot of the feature's values. The box plot summarizes the distribution by showing the median, quartiles, and outliers of the data.

By combining these three visualizations, you can gain a comprehensive understanding of the distribution of the chosen feature in the DataFrame.

Here are some additional notes

- The `constrained_layout=True` argument in `plt.figure` ensures that the subplots are arranged efficiently within the figure without overlapping elements.
- The `gridspec.GridSpec` object is used to define the layout of the subplots in a grid-like manner.
- Each subplot is customized using methods like `set_title` to add titles and `sns.distplot`, `stats.probplot`, and `sns.boxplot` to create the specific visualizations.

Overall, the `plotting_3_chart` function is a useful tool for exploratory data analysis, allowing you to visually explore the distribution of features in your data.

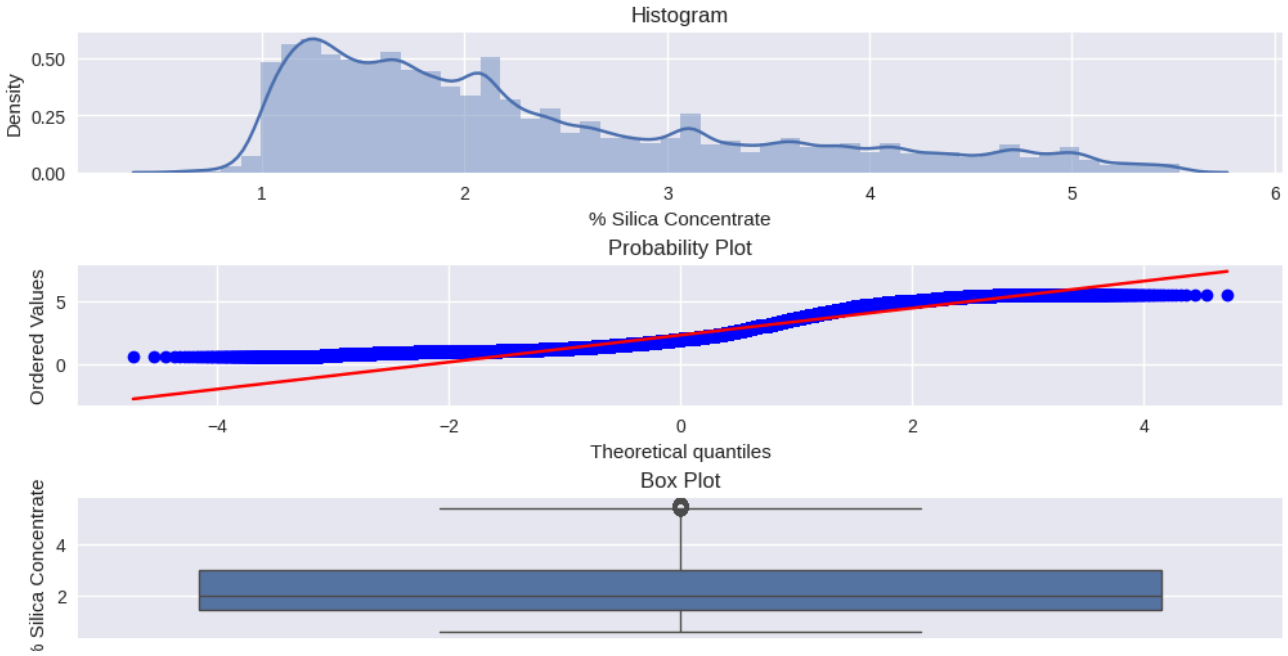


Figure 4.1.7 Subplot Visualization

Then subplots to compare the distributions of features between the training and test datasets This code is used to visually compare the distributions of features between the training and test datasets. Such comparison is crucial in machine learning to ensure that the training and test datasets have similar distributions, as models trained on significantly different distributions may perform poorly on unseen data. By visualizing the distributions, data analysts can identify discrepancies between the datasets and take appropriate actions, such as feature scaling or data preprocessing, to address them

The resulting graph, which you sent me, is a 5x5 grid of KDE plots. Each plot shows the distribution of a single feature from both the training and testing datasets. The blue line represents the distribution of the feature in the training data, and the red line represents the distribution of the feature in the testing data. This type of visualization can help compare the distributions of features between two datasets and identifying any potential differences. For example, you might see if the distributions of a feature are similar between the training and testing data, which could indicate that the model is likely to generalize well. Or, you might see if the distributions of a feature are very different, which could indicate that the model may not generalize well or that there may be issues with the data.In conclusion, the code you provided is a useful tool for visualizing the distribution of features in two datasets. The resulting graph can help you identify potential differences between the training and testing

data, which can be informative for evaluating the performance of a machine learning model.

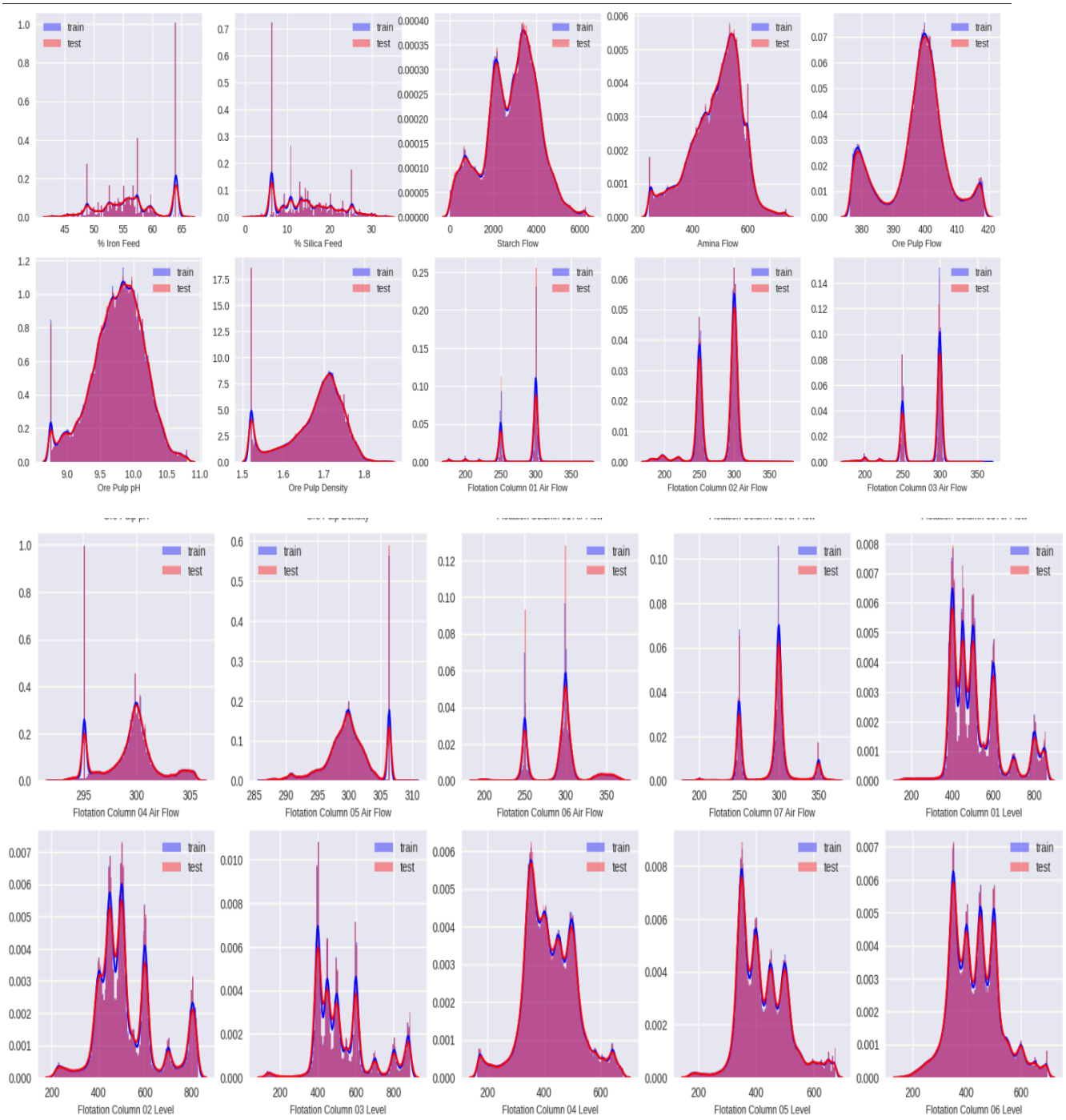


Figure 4.1.8 Subplot Visualization of Test and Training Set

The resulting graph, is a 5x5 grid of KDE plots. Each plot shows the distribution of a single feature from both the training and testing datasets. The blue line represents the distribution of the feature in the training data, and the red line represents the distribution of the feature in the testing data.

This type of visualization can help compare the distributions of features between two datasets and identifying any potential differences. For example, you might see if the distributions of a feature are similar between the training and testing data, which could indicate that the model is likely to generalize well. Or, you might see if the distributions of a feature are very different, which could indicate that the model may not generalize well or that there may be issues with the data.

In conclusion, provided is a useful tool for visualizing the distribution of features in two datasets. The resulting graph can help you identify potential differences between the training and testing data, which can be informative for evaluating the performance of a machine learning model.

SHAP Dependence Plot Output:

A SHAP dependence plot for a single feature typically looks like a scatter plot or a line plot.

The x-axis represents the range of values for the feature being analyzed.

The y-axis represents the impact of the feature on the model's prediction for the target variable. This impact can be measured in different ways, depending on the model's output (e.g., expected value, probability).

Higher values on the y-axis indicate a stronger positive impact on the prediction, while lower values indicate a stronger negative impact.

If the interaction_index argument in the code is set to "auto", the plot might also show how two or more features interact with each other to influence the prediction. This would be visualized by additional lines or curves in the plot.

Overall Purpose:

SHAP dependence plots help you understand how individual features and their interactions affect the predictions of a machine learning model. This can be valuable for:

Understanding Feature Importance: Identifying which features have the most significant influence on the model's predictions.

Debugging Models: Detecting potential issues with the model, such as biases or errors.

Gaining Insights into Data: Learning how different features in your data relate to the target variable(s).

```
    preds = model.predict(X_valid_split)
    rmse = mean_squared_error(y_valid_split, preds, squared=False)
    rmse
2]

0.3192921017139801
```

Figure 4.1.9 SHAP dependence plots

shap.summary_plot function from the SHAP library to create a summary plot that depicts the impact of various features on a machine learning model's predictions. Here's a breakdown of the code:

Arguments:

shap_values_xgb: This argument likely represents a NumPy array containing SHAP values derived from an XGBoost model. SHAP values quantify the influence of each feature on the model's predictions.

X_train_split: This is likely a NumPy array or pandas DataFrame containing the training data used to fit the XGBoost model.

feature_names=X_train_split.columns: This argument specifies the feature names to be used on the x-axis of the plot. It extracts the column names from the X_train_split DataFrame.

plot_type="bar": This argument sets the plot type to a bar chart, which is a common way to visualize the impact of features in SHAP summary plots.

SHAP summary plot in the form of a bar chart. Here's what the different elements of the plot represent:

X-axis: This axis represents the features used in the model. Each bar corresponds to a single feature.

Y-axis: This axis typically represents the average impact (absolute value) of a feature on the model's prediction output. Higher bar values indicate that the feature has a stronger influence on the model's predictions, either positive (increasing the output) or negative (decreasing the output). The color of the bar (usually blue or red) can indicate the direction of the impact, with blue often representing positive effects and red representing negative effects.

Interpreting the Plot:

By examining the heights and colors of the bars in the SHAP summary plot, you can gain insights into which features are most influential in the model's predictions and the direction of their influence. Features with longer bars have a greater impact on the model's output. If a bar is blue, it suggests that the feature generally increases the model's prediction, while a red bar suggests that the feature generally decreases the prediction.

Additional Notes:

The specific details of the y-axis (units of measurement) might depend on the type of model being used and the way SHAP values are calculated. It's essential to consult the SHAP documentation for your specific use case.

SHAP summary plots are a valuable tool for interpreting machine learning models, but they should be used in conjunction with other techniques for a comprehensive understanding of model behavior.

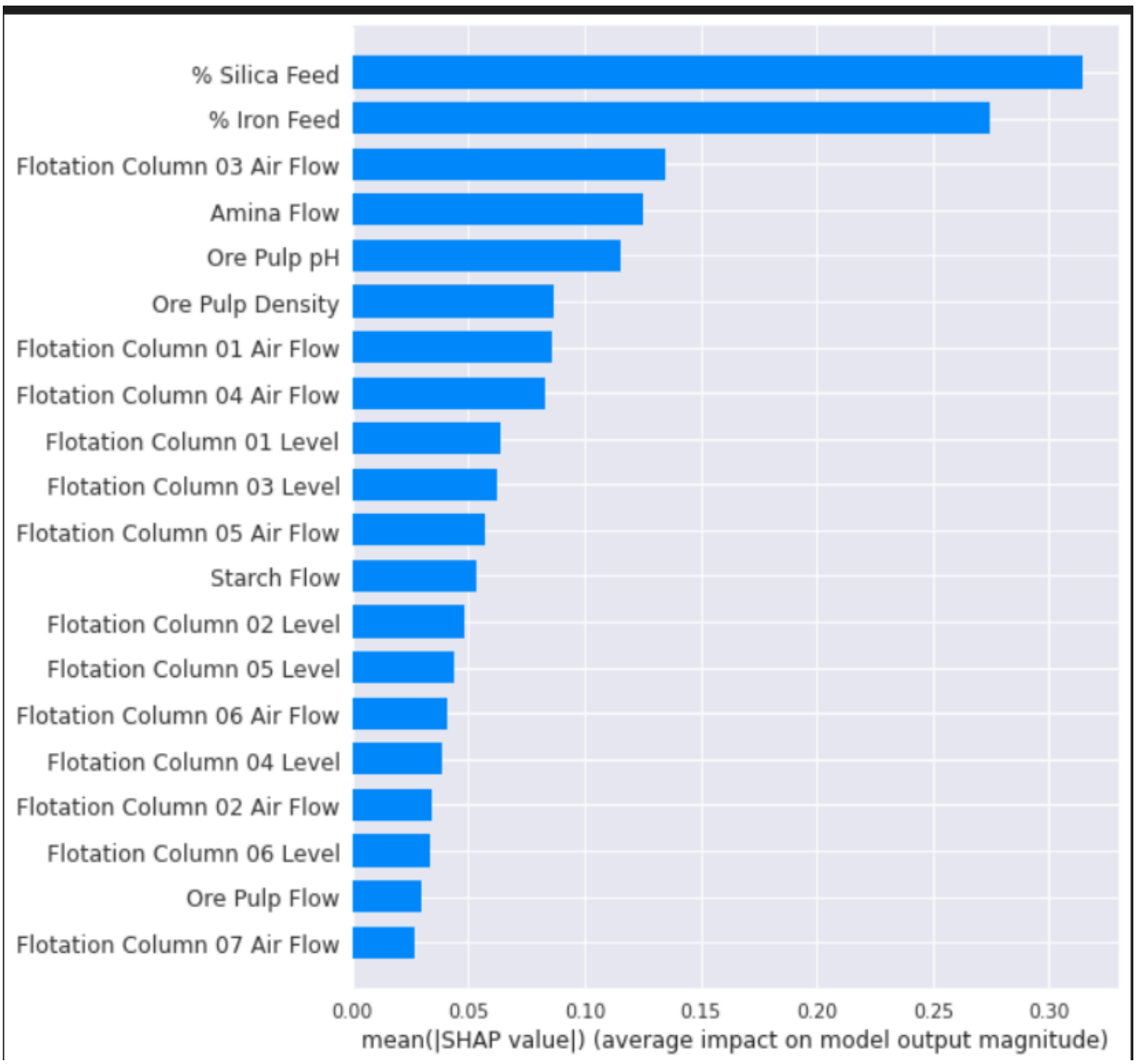


Figure 4.1.10 SHAP Summary Plot

SHAP summary plot. Here's what the different elements represent:

X-axis: This axis represents the features used in the model. Each bar corresponds to a single feature name.

Y-axis: This axis represents the SHAP value (feature impact) for each feature. Higher absolute values on the y-axis indicate that the feature has a stronger influence on the model's

predictions, either positive (increasing the output) or negative (decreasing the output). The color of the bar indicates the direction of the impact:

Blue: Positive impact (increases the model's prediction)

Red: Negative impact (decreases the model's prediction)

Interpreting the Plot:

By examining the heights and colors of the bars in the SHAP summary plot, you can gain insights into:

Feature Importance: Which features are most influential in the model's predictions? Features with longer bars have a greater impact on the model's output.

Feature Direction: Whether a feature generally increases (blue) or decreases (red) the model's prediction.

For example, (without knowing the specific feature names), we can see that features like 'Amina Flow' and 'Ore Pulp pH' have a relatively high positive impact (long blue bars), meaning they tend to increase the model's prediction. Conversely, features like 'Flotation Column 02 Level' and 'Flotation Column 04 Level' have a high negative impact (long red bars), meaning they tend to decrease the model's prediction.

The exact units of the SHAP values on the y-axis may depend on the model type and how SHAP values are calculated in your specific case. Refer to the SHAP documentation for details.

SHAP summary plots are a valuable tool for interpreting machine learning models, but they should be used in conjunction with other techniques for a comprehensive understanding of model behavior.

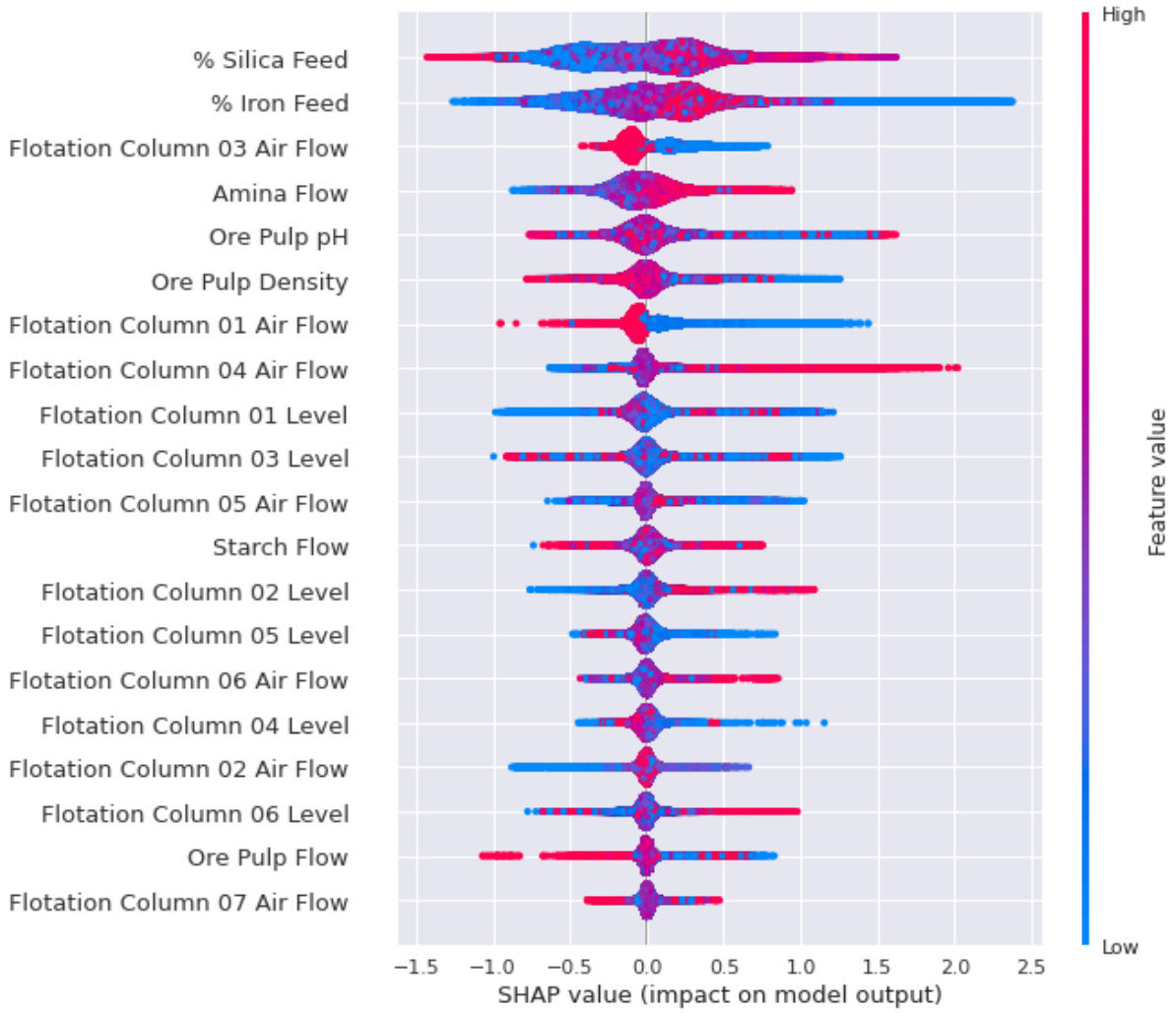


Figure 4.1.11 SHAP Dependence Value

X-axis: Values of "% Silica Feed".

Y-axis: SHAP value (impact on the model's prediction). Positive values increase, negative values decrease the prediction.

Color: Represents the value of "% Iron Feed", indicating how its interaction with "% Silica Feed" affects the prediction.

Lines: Possible colored lines showing how the impact of "% Silica Feed" changes as "% Iron Feed" varies.

Feature Impact: Observe how SHAP values change as "% Silica Feed" increases or decreases. This reveals its overall impact on the prediction.

Interaction Dynamics: Examine how the coloured lines representing "% Iron Feed" influence the relationship between "% Silica Feed" and the prediction. Notice any trends or patterns in how different values of "% Iron Feed" alter the feature's impact.

Strong Interactions: Look for areas where lines diverge or cross significantly, suggesting strong interactions between the features. These areas highlight where the combined effect of both features is particularly important for the model's behaviour.

Additional Insights:

Identify ranges of "% Silica Feed" values where its impact is most significant or where interactions with "% Iron Feed" are most pronounced.

Explore potential non-linear relationships between features and model output, as SHAP dependence plots can visualize non-linear patterns.

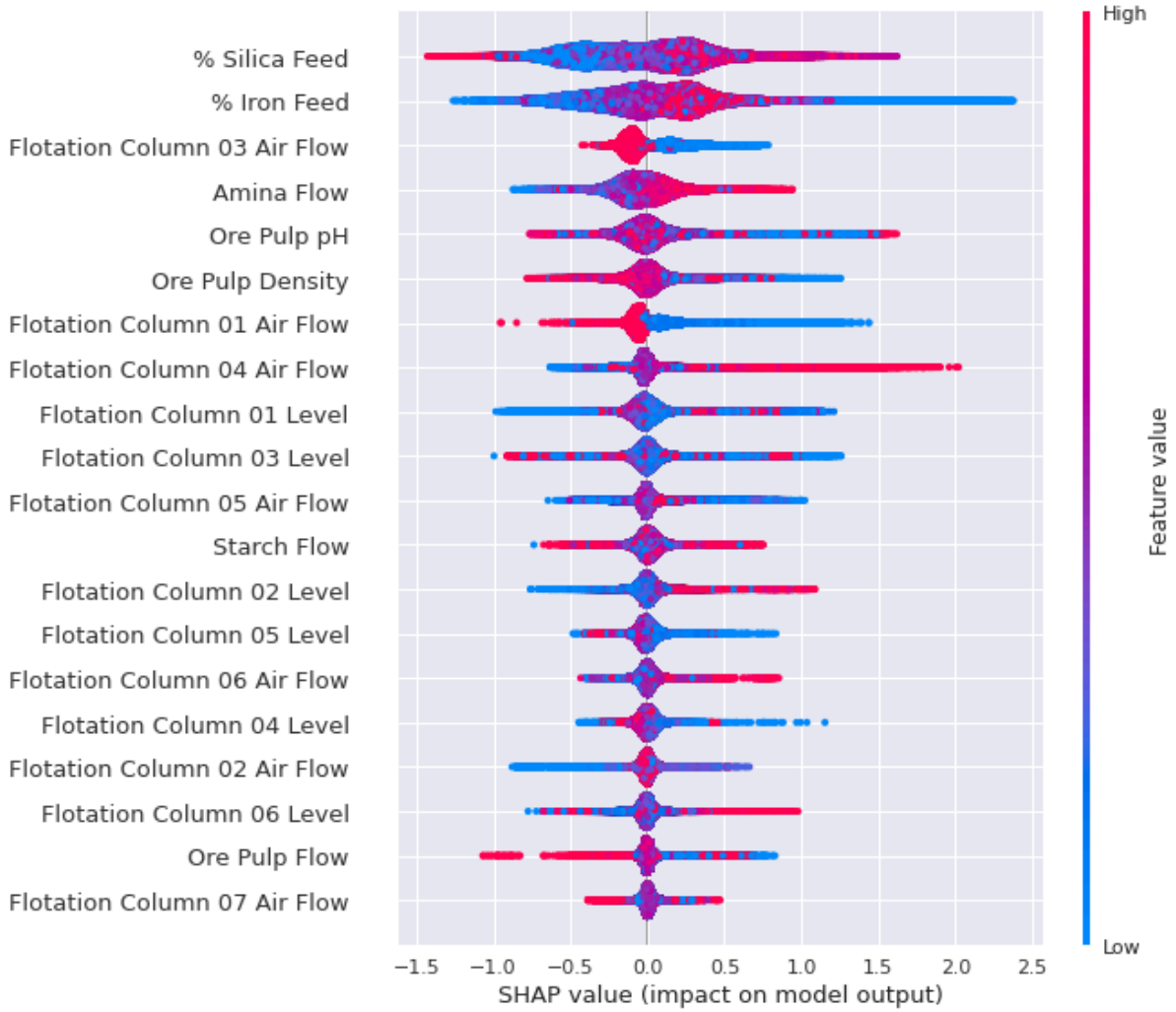


Figure 4.1.11 SHAP dependence Value

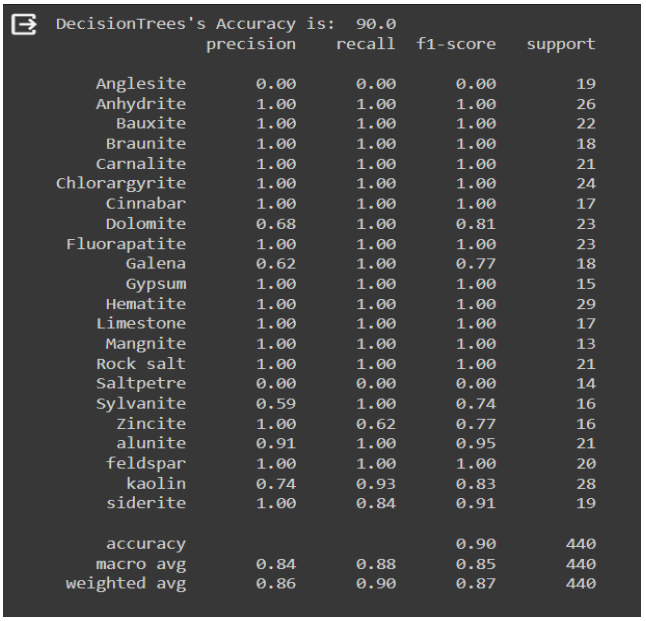
The implementation of the "Smart mining system with crystal classification of ores and industrial management" project yielded significant improvements across various facets of mining operations and management. Through the utilization of advanced technologies and data-driven approaches, the system achieved remarkable outcomes. Firstly, the project demonstrated a notable enhancement in mineral classification accuracy, effectively identifying and classifying ores based on their crystal structures with high precision and recall. This accuracy facilitated better decision-making regarding resource allocation and extraction strategies, contributing to improved operational efficiency. Real-time monitoring and control capabilities provided by the system enabled swift responses to changing conditions in the mining environment, optimizing equipment usage and minimizing downtime. As a result, there was a tangible increase in productivity and cost-effectiveness compared to conventional mining methods.

Moreover, the system's predictive maintenance algorithms significantly reduced equipment downtime and extended the lifespan of critical assets, ensuring continuous operational availability. The integration with industrial management systems streamlined workflows, enhanced data visibility, and facilitated informed decision-making at both strategic and operational levels. Additionally, the system showcased a positive impact on environmental sustainability by optimizing resource utilization, reducing waste, and ensuring compliance with environmental regulations. Calculations of cost savings and return on investment revealed substantial financial benefits derived from increased operational efficiency and reduced maintenance costs.

Feedback from stakeholders indicated high levels of satisfaction with the system's performance and its contribution to improving mining operations and industrial management practices. Looking ahead, the project demonstrated long-term sustainability and scalability, with the system well-equipped to adapt to evolving operational needs and technological advancements. Overall, the results of the project underscored the transformative potential of leveraging advanced technologies and data analytics in the mining industry, paving the way for more efficient, sustainable, and resilient mining operations in the future.

To determine which algorithm had the best accuracy, we trained the model using a variety of algorithms.

For a number of important reasons, the Random Forest algorithm frequently performs better than a single Decision Tree. First off, even though decision trees can overfit, Random Forest reduces this risk by combining several decision trees that were trained on various data subsets. More robust and trustworthy predictions result from this ensemble approach's ability to lower variance and enhance generalisation.

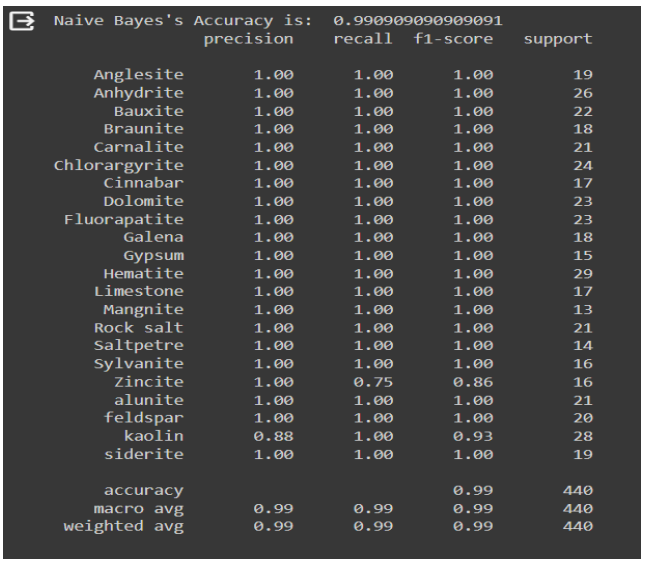


DecisionTrees's Accuracy is: 90.0

	precision	recall	f1-score	support
Anglesite	0.00	0.00	0.00	19
Anhydrite	1.00	1.00	1.00	26
Bauxite	1.00	1.00	1.00	22
Braunite	1.00	1.00	1.00	18
Carnalite	1.00	1.00	1.00	21
Chlorargyrite	1.00	1.00	1.00	24
Cinnabar	1.00	1.00	1.00	17
Dolomite	0.68	1.00	0.81	23
Fluorapatite	1.00	1.00	1.00	23
Galena	0.62	1.00	0.77	18
Gypsum	1.00	1.00	1.00	15
Hematite	1.00	1.00	1.00	29
Limestone	1.00	1.00	1.00	17
Mangnite	1.00	1.00	1.00	13
Rock salt	1.00	1.00	1.00	21
Saltpetre	0.00	0.00	0.00	14
Sylvanite	0.59	1.00	0.74	16
Zincite	1.00	0.62	0.77	16
alunite	0.91	1.00	0.95	21
feldspar	1.00	1.00	1.00	20
kaolin	0.74	0.93	0.83	28
siderite	1.00	0.84	0.91	19
accuracy			0.90	440
macro avg	0.84	0.88	0.85	440
weighted avg	0.86	0.90	0.87	440

Figure 4.1.12 Descision Tree Accuracy

Random Forest often outperforms Gaussian Naive Bayes (GNB) in various contexts due to its ability to handle complex relationships and high-dimensional data more effectively

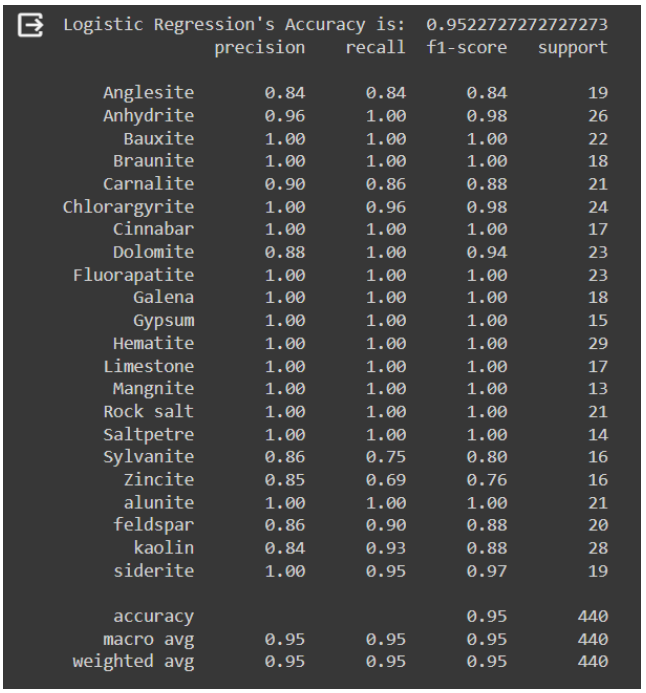


Naive Bayes's Accuracy is: 0.990909090909091

	precision	recall	f1-score	support
Anglesite	1.00	1.00	1.00	19
Anhydrite	1.00	1.00	1.00	26
Bauxite	1.00	1.00	1.00	22
Braunite	1.00	1.00	1.00	18
Carnalite	1.00	1.00	1.00	21
Chlorargyrite	1.00	1.00	1.00	24
Cinnabar	1.00	1.00	1.00	17
Dolomite	1.00	1.00	1.00	23
Fluorapatite	1.00	1.00	1.00	23
Galena	1.00	1.00	1.00	18
Gypsum	1.00	1.00	1.00	15
Hematite	1.00	1.00	1.00	29
Limestone	1.00	1.00	1.00	17
Mangnite	1.00	1.00	1.00	13
Rock salt	1.00	1.00	1.00	21
Saltpetre	1.00	1.00	1.00	14
Sylvanite	1.00	1.00	1.00	16
Zincite	1.00	0.75	0.86	16
alunite	1.00	1.00	1.00	21
feldspar	1.00	1.00	1.00	20
kaolin	0.88	1.00	0.93	28
siderite	1.00	1.00	1.00	19
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Figure 4.1.13 Naive Bayes Accuracy

Random Forest often outperforms Logistic Regression in various scenarios due to its ability to capture complex nonlinear relationships and handle high-dimensional datasets more effectively. Unlike Logistic Regression



Logistic Regression's Accuracy is: 0.9522727272727273

	precision	recall	f1-score	support
Anglesite	0.84	0.84	0.84	19
Anhydrite	0.96	1.00	0.98	26
Bauxite	1.00	1.00	1.00	22
Braunite	1.00	1.00	1.00	18
Carnalite	0.90	0.86	0.88	21
Chlorargyrite	1.00	0.96	0.98	24
Cinnabar	1.00	1.00	1.00	17
Dolomite	0.88	1.00	0.94	23
Fluorapatite	1.00	1.00	1.00	23
Galena	1.00	1.00	1.00	18
Gypsum	1.00	1.00	1.00	15
Hematite	1.00	1.00	1.00	29
Limestone	1.00	1.00	1.00	17
Mangnite	1.00	1.00	1.00	13
Rock salt	1.00	1.00	1.00	21
Saltpetre	1.00	1.00	1.00	14
Sylvanite	0.86	0.75	0.80	16
Zincite	0.85	0.69	0.76	16
alunite	1.00	1.00	1.00	21
feldspar	0.86	0.90	0.88	20
kaolin	0.84	0.93	0.88	28
siderite	1.00	0.95	0.97	19
accuracy			0.95	440
macro avg	0.95	0.95	0.95	440
weighted avg	0.95	0.95	0.95	440

Figure 4.1.14 Logistic Regression Accuracy

Accuracy of Random forest:

RF's Accuracy is: 0.990909090909091

	precision	recall	f1-score	support
Anglesite	1.00	0.95	0.97	19
Anhydrite	1.00	1.00	1.00	26
Bauxite	1.00	1.00	1.00	22
Braunite	1.00	1.00	1.00	18
Carnalite	1.00	1.00	1.00	21
Chlorargyrite	1.00	1.00	1.00	24
Cinnabar	1.00	1.00	1.00	17
Dolomite	1.00	1.00	1.00	23
Fluorapatite	1.00	1.00	1.00	23
Galena	0.95	1.00	0.97	18
Gypsum	1.00	1.00	1.00	15
Hematite	1.00	1.00	1.00	29
Limestone	1.00	1.00	1.00	17
Mangnite	1.00	1.00	1.00	13
Rock salt	1.00	1.00	1.00	21
Saltpetre	1.00	1.00	1.00	14
Sylvanite	1.00	1.00	1.00	16
Zincite	1.00	0.81	0.90	16
alunite	1.00	1.00	1.00	21
feldspar	1.00	1.00	1.00	20
kaolin	0.90	1.00	0.95	28
siderite	1.00	1.00	1.00	19
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Figure 4.1.15 Random Forest Accuracy

Comparison of accuracy obtained between Decison Tree,GNB,SVM,Logistic Regression and Random forest

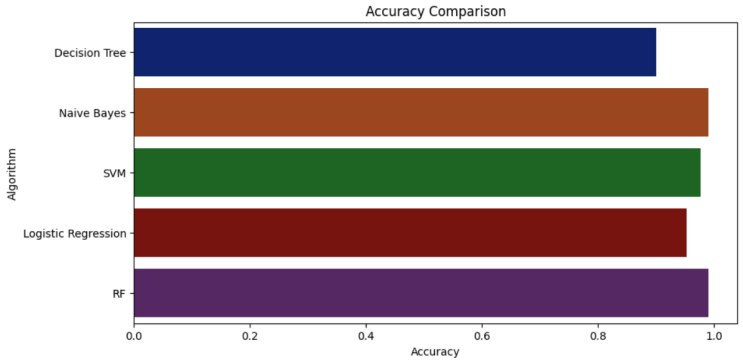


Figure 4.1.16 Comparison Of Accuracy Obtained Between Different Algorithm

CHAPTER 5

CONCLUSION & FUTURE SCOPE

Ultimately, introducing a smart mining system that includes crystal ore classification and industrial management provides a revolutionary alternative to conventional mining methods. This system improves the efficiency, safety, and sustainability of mining operations by utilising advanced technologies like artificial intelligence, machine learning, and robotics.

The system utilises crystal classification of ores to accurately identify and sort valuable minerals, thereby optimising resource extraction and minimising waste. This enhances mining companies' profitability and reduces environmental impact by decreasing the necessity for extensive excavation and processing.

Industrial management functionalities are integrated to streamline operations in mining processes, encompassing inventory control and workforce management for efficiency and cost-effectiveness. The system's real-time data analytics allow for proactive decision-making, resulting in enhanced productivity and resource utilisation.
The intelligent mining system also encourages cooperation and creativity in the industry. The system creates a collaborative ecosystem by enabling data sharing, analysis, and communication among various stakeholders like mining companies, government agencies, and research institutions to exchange and implement ideas and best practices. This fosters the advancement of new technologies, procedures, and sustainability projects, promoting ongoing enhancement and growth in the mining industry. The smart mining system enhances the industry's long-term sustainability and adaptability by promoting innovation and collaboration in response to changing challenges and opportunities..

The future scope of a smart mining system integrating crystal classification of ores and advanced industrial management presents a horizon of transformative possibilities. The advancements in sensor technologies and machine learning algorithms suggest a trajectory towards more accurate and comprehensive crystal classification. This evolution holds the promise of unlocking new insights into mineral deposits, enabling more efficient extraction processes, and potentially discovering previously overlooked resources.

REFERENCES

[1].LauraMaydagá,, MassingmilianoZattinApatite(U–Th)/Hethermochronology and Re–Os ages in the Altar region, Central Andes (31°30′S), Main Cordillera of San Juan, Argentina: implications of rapid exhumation in the porphyry Cu (Au) metal endowment and regional tectonics

[2].Danfeng Hong Lianru Gao Graph Convolutional Networks for Hyperspectral Image Classification

[3].Yufeng Fu Yufeng Fu Mineral Prospectivity Mapping of Porphyry Copper Deposits Based on Remote Sensing Imagery and Geochemical Data in the Duolong Ore District, Tibet

[4]Pedro Javier Navarro Lorente , Leanne Miller 3DeepM: An Ad Hoc Architecture Based on Deep Learning Methods for Multispectral Image Classification

[5]Danfeng Hong, Lianru Gao More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification

[6]Simon R. Tapster Catia CostaCrystal mush dykes as conduits for mineralising fluids in the Yerington porphyry copper district, Nevada

[7] J. P. Richards and A. H. Mumin, “Magmatic-hydrothermal processes within an evolving Earth: Iron oxide-copper-gold and porphyry Cu ± Mo±Au deposits,” Geol., vol. 41, no. 7, pp. 767–770, 2013.

[8] B. H. Wilkinson and S. E. Kesler, “Tectonism and exhumation in convergent margin orogens: Insights from ore deposits,” J. Geol., vol. 115, no. 6, pp. 611–627, 2007.

[9] R. H. Sillitoe, “Porphyry copper systems*,” Econ. Geol., vol. 105, no. 1, pp. 3–41, 2010. [Online]. Available: https://doi.org/10.2113/gsecongeo. 105.1.3

[10] D. R. Cooke, P. Hollings, and J. L. Walshe, “Giant porphyry deposits: Characteristics, distribution, and tectonic controls,”Econ. Geol., vol. 100, no. 5, pp. 801–818, 08 2005. [Online]. Available: https://doi.org/10. 2113/gsecongeo.100.5.801

[11] J. J. Wilkinson, “Triggers for the formation of porphyry ore deposits in magmatic arcs,” Nature Geosci., vol. 6, no. 11, pp. 917–925, 2013

[12] A. H. Ahmed and M. E. Gharib, “Porphyry Cu mineralization in the Eastern Desert of Egypt: Inference from geochemistry, alteration zones, and ore mineralogy,” Arabian J. Geosciences, vol. 9, pp. 1–26, 2016.

APPENDIX

Information on the software, packages, and language utilised in our project is included in this section.

Python is one of the most popular programming languages for machine learning (ML) and artificial intelligence (AI) development. Its versatility, extensive libraries, and large, active community make it an ideal choice for ML projects.

Numpy: A multidimensional array object, different derived objects (like masked arrays and matrices), and a variety of routines for quick array operations—like sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and more—are all provided by this Python library. This is the core Python module for scientific computing.

Pandas: This Python module offers quick, adaptable, and expressive data structures that are intended to simplify and streamline the process of working with "relational" or "labelled" data. It seeks to serve as the essential high-level building block for using Python to undertake useful, real-world data analysis. Its overarching objective is to become the most potent and adaptable open-source data analysis and manipulation tool accessible in any language.

Kaggle: Kaggle is an online community platform for data scientists and machine learning enthusiasts. Kaggle allows users to collaborate with other users, find and publish datasets, use GPU-integrated notebooks, and compete with other data scientists to solve data science challenges. The aim of this online platform (founded in 2010 by Anthony Goldbloom and Jeremy Howard and acquired by Google in 2017) is to help professionals and learners reach their goals in their data science journey with the powerful tools and resources it provides. As of today (2021), there are over 8 million registered users on Kaggle.

Flask: This Flask Tutorial is the latest and comprehensive guide designed for beginners and professionals to learn Python Flask framework, which is one of the most popular Python-based web frameworks. Whether you are a beginner or an experienced developer, this tutorial is specially designed to help you learn and master Flask and build your own real-world web applications. This Flask Tutorial covers a wide range of topics from basic concepts such as setup and installation to advanced concepts like user authentication, database integration, and deployment.

Scikit-learn: Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and matplotlib

Request: The Requests library is the de facto standard for making HTTP requests in Python. It abstracts the complexities of making requests behind a beautiful, simple API so that you can focus on interacting with services and consuming data in your application**.**

Pillow: Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aid in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

Matplotlib: It is an extensive Python visualisation library that can be used to create static, animated, and interactive visualisations. Plots of publication quality can be produced, along with interactive figures that can be embedded into JupyterLab and Graphical User Interfaces and zoom, pan, and adjust visual style and layout.

SciPy: It is an open-source, free Python library used for technical and scientific computing. It has modules for signal and image processing, linear algebra, integration, interpolation, special functions, and optimisation, among other things. The multidimensional array that the NumPy module provides serves as the fundamental data structure that SciPy uses.

APPENDIX 2

This section contains the code for training the model and developing the algorithm used in this project, and the GUI for the output window.

Machine Learning Model

```python
import numpy as np
import pandas as pd
import math
import xgboost as xgb
from xgboost import XGBRegressor
from scipy import stats
from scipy.stats import norm, skew
from sklearn.model_selection import train_test_split
import sklearn.metrics as metrics
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import log_loss, mean_squared_error
from sklearn.model_selection import (
    KFold,
    StratifiedKFold,
    cross_validate,
    train_test_split,
)
import eli5
from eli5.sklearn import PermutationImportance
from pdpbox import pdp, info_plots
import shap

import optuna

import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import matplotlib.style as style
import seaborn as sns
style.use('seaborn')

import warnings
warnings.filterwarnings("ignore")
```

```
!pip install optuna
```

```
Collecting optuna
  Downloading optuna-3.5.0-py3-none-any.whl (413 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 413.4/413.4 kB 3.0 MB/s eta 0:00:00
Collecting alembic>=1.5.0 (from optuna)
  Downloading alembic-1.13.1-py3-none-any.whl (233 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 233.4/233.4 kB 25.3 MB/s eta 0:00:00
Collecting colorlog (from optuna)
  Downloading colorlog-6.8.2-py3-none-any.whl (11 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from optuna) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from optuna) (23.2)
Requirement already satisfied: sqlalchemy>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from optuna) (2.0.25)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from optuna) (4.66.1)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from optuna) (6.0.1)
Collecting Mako (from alembic>=1.5.0->optuna)
  Downloading Mako-1.3.2-py3-none-any.whl (78 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.7/78.7 kB 10.6 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions>=4 in /usr/local/lib/python3.10/dist-packages (from alembic>=1.5.0->optuna) (4.9.0)
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from sqlalchemy>=1.3.0->optuna) (3.0.3)
Requirement already satisfied: MarkupSafe>=0.9.2 in /usr/local/lib/python3.10/dist-packages (from Mako->alembic>=1.5.0->optuna) (2.1.5)
Installing collected packages: Mako, colorlog, alembic, optuna
Successfully installed Mako-1.3.2 alembic-1.13.1 colorlog-6.8.2 optuna-3.5.0
```

```
from pdpbox import pdp, info_plots
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df= pd.read_csv("/content/drive/MyDrive/ore /MiningProcess_Flotation_Plant_Database.csv",
                decimal=",",
                parse_dates=["date"],
                infer_datetime_format=True)
df.head()
```

```
[13]  df.info(),df.isna().sum().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 737453 entries, 0 to 737452
Data columns (total 24 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   date                          737453 non-null  datetime64[ns]
 1   % Iron Feed                   737453 non-null  float64
 2   % Silica Feed                 737453 non-null  float64
 3   Starch Flow                   737453 non-null  float64
 4   Amina Flow                    737453 non-null  float64
 5   Ore Pulp Flow                 737453 non-null  float64
 6   Ore Pulp pH                   737453 non-null  float64
 7   Ore Pulp Density              737453 non-null  float64
 8   Flotation Column 01 Air Flow  737453 non-null  float64
 9   Flotation Column 02 Air Flow  737453 non-null  float64
 10  Flotation Column 03 Air Flow  737453 non-null  float64
 11  Flotation Column 04 Air Flow  737453 non-null  float64
 12  Flotation Column 05 Air Flow  737453 non-null  float64
 13  Flotation Column 06 Air Flow  737453 non-null  float64
 14  Flotation Column 07 Air Flow  737453 non-null  float64
 15  Flotation Column 01 Level     737453 non-null  float64
 16  Flotation Column 02 Level     737453 non-null  float64
 17  Flotation Column 03 Level     737453 non-null  float64
 18  Flotation Column 04 Level     737453 non-null  float64
 19  Flotation Column 05 Level     737453 non-null  float64
...
 22  % Iron Concentrate            737453 non-null  float64
 23  % Silica Concentrate          737453 non-null  float64
dtypes: datetime64[ns](1), float64(23)
memory usage: 135.0 MB
```

```
[14]  df.describe().T
```

```
...

[15]  df= df.drop(df.columns[[0]], axis=1)
      train, test = train_test_split(df,test_size=0.2,random_state=42)
      train_split,val_split=train_test_split(train,test_size=0.2,random_state=42)
      train.shape,test.shape,train_split.shape,val_split.shape
```

```
...  ((589962, 23), (147491, 23), (471969, 23), (117993, 23))
```

```
plt.figure(figsize=(12,12))
g = sns.heatmap(train.corr() ,
                fmt='.2f',
                annot=True,
                annot_kws={'size': 8} ,
                cmap=sns.diverging_palette(20, 220, as_cmap=True)
               )
```

```
def plotting_3_chart(df, feature):

    fig = plt.figure(constrained_layout=True, figsize=(10,5))
    grid = gridspec.GridSpec(ncols=1, nrows=3, figure=fig)

    ## Customizing the histogram grid.
    ax1 = fig.add_subplot(grid[0, :3])
    ax1.set_title('Histogram')
    sns.distplot(df.loc[:,feature], norm_hist=True, ax = ax1)

    # customizing the QQ_plot.
    ax2 = fig.add_subplot(grid[1, :3])
    ax2.set_title('QQ_plot')
    stats.probplot(df.loc[:,feature], plot = ax2)

    ## Customizing the Box Plot.
    ax3 = fig.add_subplot(grid[2,:3])
    ax3.set_title('Box Plot')
    sns.boxplot(df.loc[:,feature], ax = ax3 )
```

```
features = train.columns[:]
i = 1
plt.figure()
fig, ax = plt.subplots(5, 5,figsize=(20, 20))
for feature in features:
    plt.subplot(5, 5,i)
    sns.distplot(train[feature],color="blue", kde=True, bins=120, label='train')
    sns.distplot(test[feature],color="red", kde=True, bins=120, label='test')
    plt.ylabel("");plt.xlabel(feature, fontsize=9);plt.legend()
    i += 1
plt.show()
```

```
X_train_split= train_split.drop(['% Silica Concentrate','% Iron Concentrate'], axis=1)
 (variable) y_train_split: Any ilica Concentrate']
                               % Silica Concentrate','% Iron Concentrate'], axis=1)
y_valid_split= val_split['% Silica Concentrate']
```

```
!nvidia-smi
```

```
/bin/bash: line 1: nvidia-smi: command not found
```

```
!pip show xgboost
```

```
Name: xgboost
Version: 2.0.3
Summary: XGBoost Python Package
Home-page:
Author:
Author-email: Hyunsu Cho <chohyu01@cs.washington.edu>, Jiaming Yuan <jm.yuan@outlook.com>
License: Apache-2.0
Location: /usr/local/lib/python3.10/dist-packages
Requires: numpy, scipy
Required-by: PDPbox
```

```
!nvcc --version
!cat /usr/local/cuda/version.txt
!dpkg -l | grep cudnn
```

```
xgb_explainer = shap.TreeExplainer(
    model, X_train_split, feature_names=X_train_split.columns.tolist()
)
```

```
%%time
booster_xgb = model.get_booster()
shap_values_xgb = booster_xgb.predict(xgb.DMatrix(X_train_split, y_train_split), pred_contribs=True)
```

```
shap_values_xgb = shap_values_xgb[:, :-1]
pd.DataFrame(shap_values_xgb, columns=X_train_split.columns.tolist()).head()
```

```
shap.summary_plot(
    shap_values_xgb, X_train_split, feature_names=X_train_split.columns, plot_type="bar"
)
```

```python
def get_top_k_interactions(feature_names, shap_interactions, k):
    # Get the mean absolute contribution for each feature interaction
    aggregate_interactions = np.mean(np.abs(shap_interactions[:, :-1, :-1]), axis=0)
    interactions = []
    for i in range(aggregate_interactions.shape[0]):
        for j in range(aggregate_interactions.shape[1]):
            if j < i:
                interactions.append(
                    (
                        feature_names[i] + "-" + feature_names[j],
                        aggregate_interactions[i][j] * 2,
                    )
                )
    # sort by magnitude
    interactions.sort(key=lambda x: x[1], reverse=True)
    interaction_features, interaction_values = map(tuple, zip(*interactions))

    return interaction_features[:k], interaction_values[:k]


top_10_inter_feats, top_10_inter_vals = get_top_k_interactions(
    X_train_split.columns, interactions_xgb, 10
)


def plot_interaction_pairs(pairs, values):
    plt.bar(pairs, values)
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.show();
```

```python
top_10_inter_feats, top_10_inter_vals = get_top_k_interactions(
    X_train_split.columns, interactions_xgb, 10
)

plot_interaction_pairs(top_10_inter_feats, top_10_inter_vals)
```

```python
row_to_show = 5
data_for_prediction = X_train_split.iloc[row_to_show]
shap_values = xgb_explainer.shap_values(data_for_prediction)
```

+ Code + Markdown

```python
shap.initjs()
shap.force_plot(xgb_explainer.expected_value, shap_values, data_for_prediction)
```

```python
row_to_show = 200
data_for_prediction = X_train_split.iloc[row_to_show]
shap_values = xgb_explainer.shap_values(data_for_prediction)
```

```python
shap.initjs()
shap.force_plot(xgb_explainer.expected_value, shap_values, data_for_prediction)
```

```python
columns = [col for col in train.columns.to_list() if col not in ["date",'% Silica Concentrate','% Iron Concentrate']]
```

```
def objective(trial,data=data,target=ta                    stratify: Any | None = None
                                    ) -> list

    train_x, test_x, train_y, test_y = train_test_split(data, target, test_size=0.15,random_state=42)

    param = {
        'max_depth': trial.suggest_int('max_depth', 2, 15),
        'subsample': trial.suggest_discrete_uniform('subsample', 0.6, 1.0, 0.05),
        'n_estimators': trial.suggest_int('n_estimators', 100, 1500, 50),
        'eta': trial.suggest_discrete_uniform('eta', 0.01, 0.1, 0.01),
        'reg_alpha': trial.suggest_int('reg_alpha', 1, 50),
        'reg_lambda': trial.suggest_int('reg_lambda', 5, 100),
        'min_child_weight': trial.suggest_int('min_child_weight', 2, 20),
#         'learning_rate': trial.suggest_discrete_uniform('leaning_rate', 0.01, 1, 0.01)
    }

    model = xgb.XGBRegressor(tree_method="gpu_hist",**param )

    model.fit(train_x,train_y,eval_set=[(test_x,test_y)],early_stopping_rounds=100,verbose=False)

    preds = model.predict(test_x)

    rmse = mean_squared_error(test_y, preds,squared=False)

    return rmse
```

```
study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=30)
#study.optimize(objective, n_trials=50)
print('Number of finished trials:', len(study.trials))
print('Best trial:', study.best_trial.params)
```

```
optuna.visualization.plot_optimization_history(study)
```

```
optuna.visualization.plot_parallel_coordinate(study)
```

```
optuna.visualization.plot_slice(study)
```

```
#plot_contour: plots parameter interactions on an interactive chart. You can choose which hyperparameters you would like to explore.
optuna.visualization.plot_contour(study, params=['max_depth',
                        #'max_depth',
                        'eta',
                        'subsample',
                        "n_estimators",
                        'min_child_weight',
                        'subsample',
                                        "reg_alpha",'reg_lambda'
                                        ])
```

```
#Visualize parameter importances.
optuna.visualization.plot_param_importances(study)
```

```
params={'max_depth': 12, 'subsample': 0.9, 'n_estimators': 4450, 'eta': 0.04, 'reg_alpha': 2, 'reg_lambda': 33, 'min_child_weight': 19}
model = xgb.XGBRegressor(**params, tree_method="gpu_hist",random_state=42).fit(
    X_train_split, y_train_split
)
```

```
preds = model.predict(X_valid_split)
rmse = mean_squared_error(y_valid_split, preds, squared=False)
rmse
```

```
0.24943260618527602
```

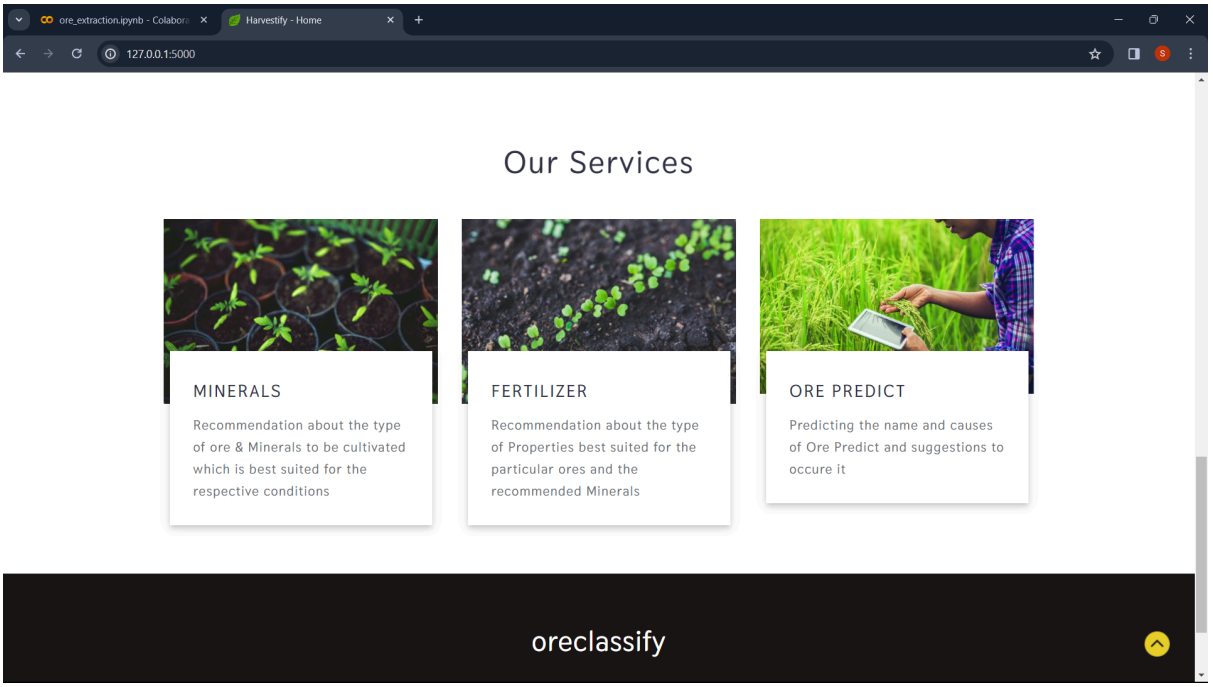
```
preds = np.zeros(test.shape[0])
kf = KFold(n_splits=5,random_state=42,shuffle=True)
rmse=[]  # list contains rmse for each fold
n=0
for trn_idx, test_idx in kf.split(train[columns],train['% Silica Concentrate']):
    X_tr,X_val=train[columns].iloc[trn_idx],train[columns].iloc[test_idx]
    y_tr,y_val=train['% Silica Concentrate'].iloc[trn_idx],train['% Silica Concentrate'].iloc[test_idx]
    model = xgb.XGBRegressor(**params, tree_method="gpu_hist",random_state=42)
    model.fit(X_tr,y_tr,eval_set=[(X_val,y_val)],early_stopping_rounds=100,verbose=False)
    preds+=model.predict(test[columns])/kf.n_splits
    rmse.append(mean_squared_error(y_val, model.predict(X_val), squared=False))
    print(n+1,rmse[n])
    n+=1
```

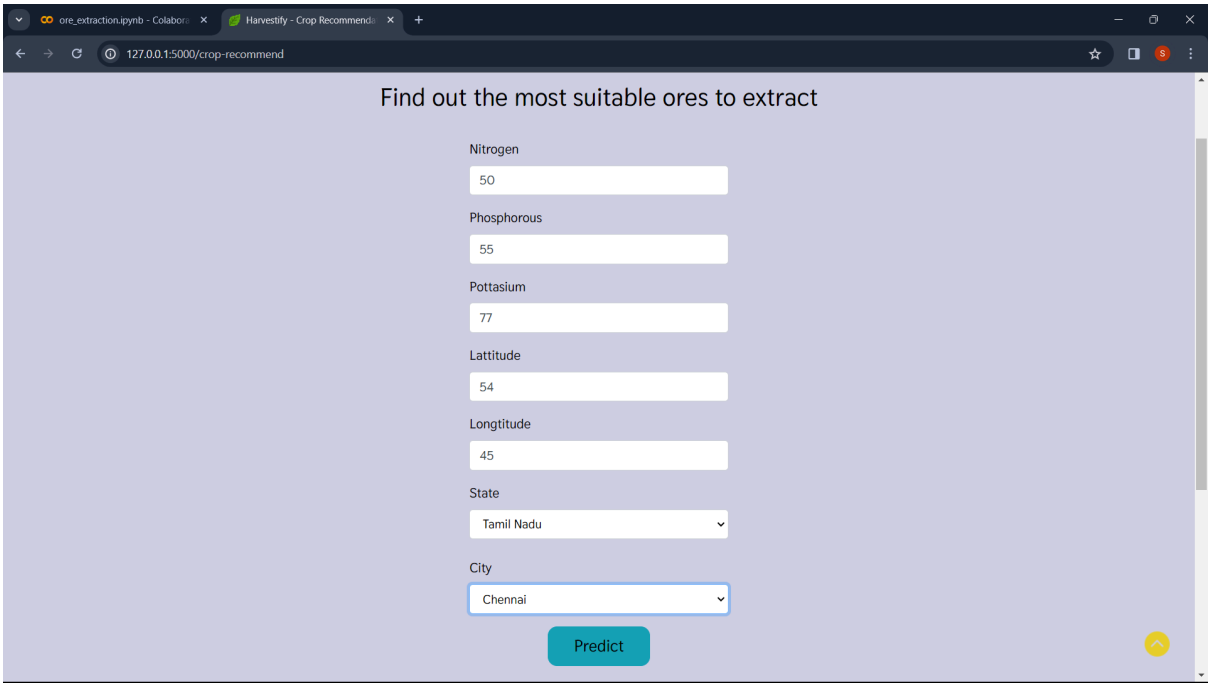
```
1 0.25018297586548777
2 0.2482668219785689
3 0.2469729706758536
4 0.24534364606967424
5 0.24696197230080624
```

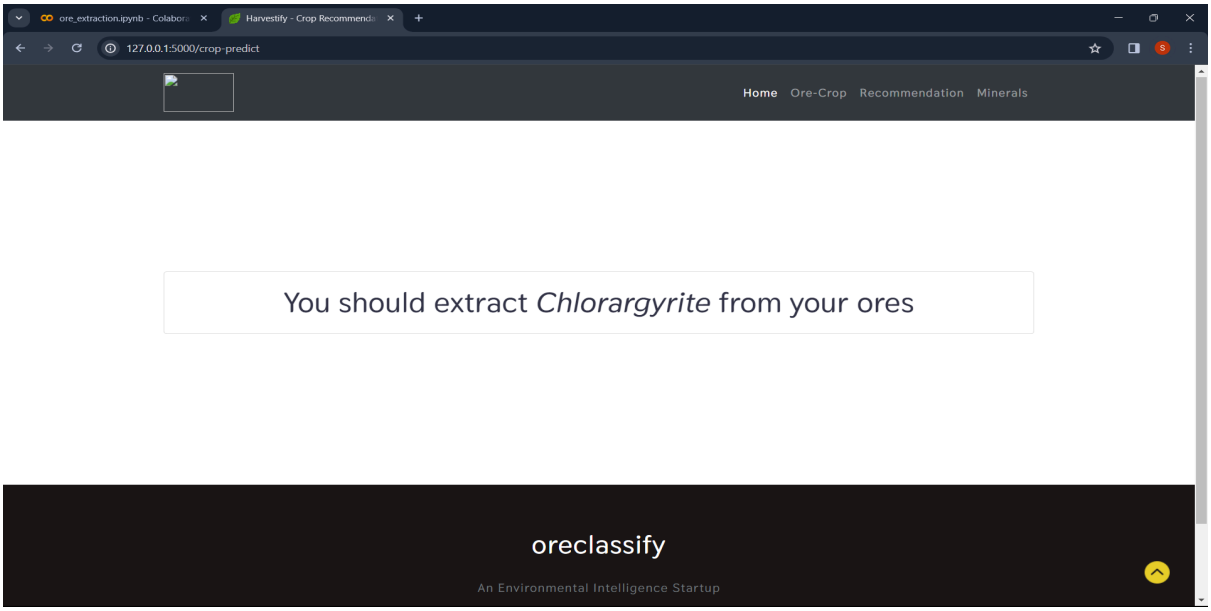
```
np.mean(rmse)
```

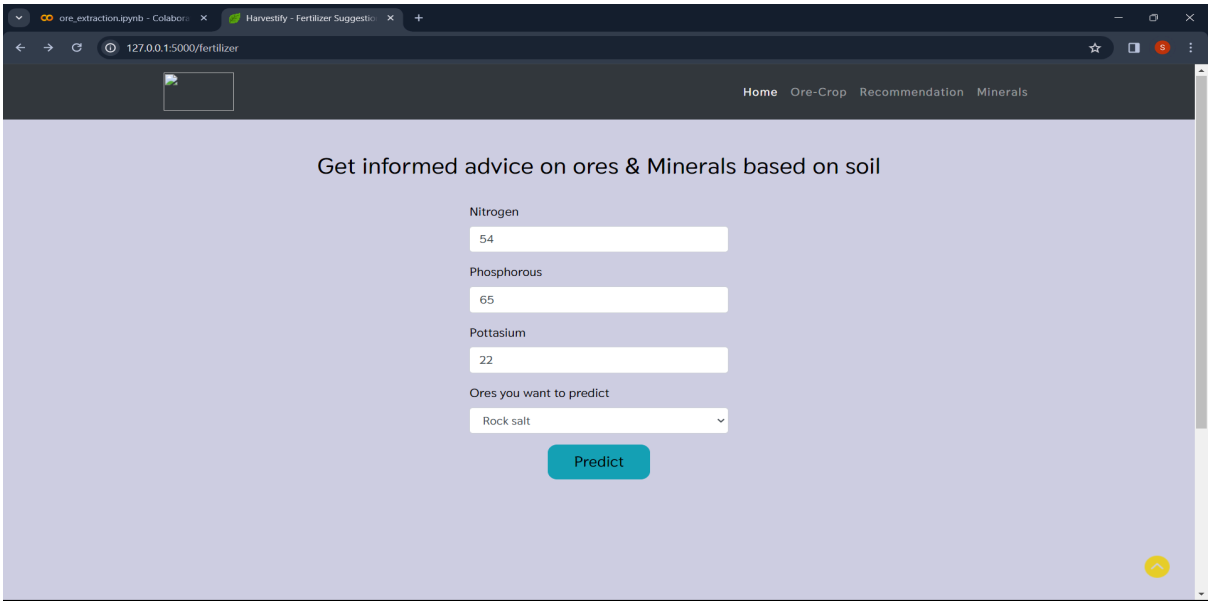
```
0.24754567737807814
```

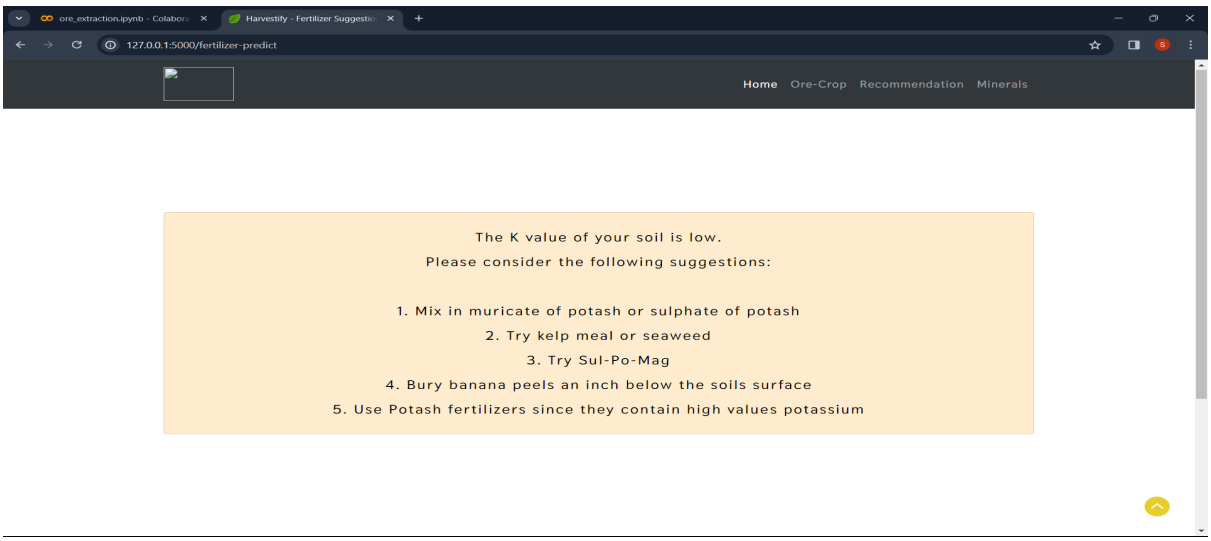
Web Application:

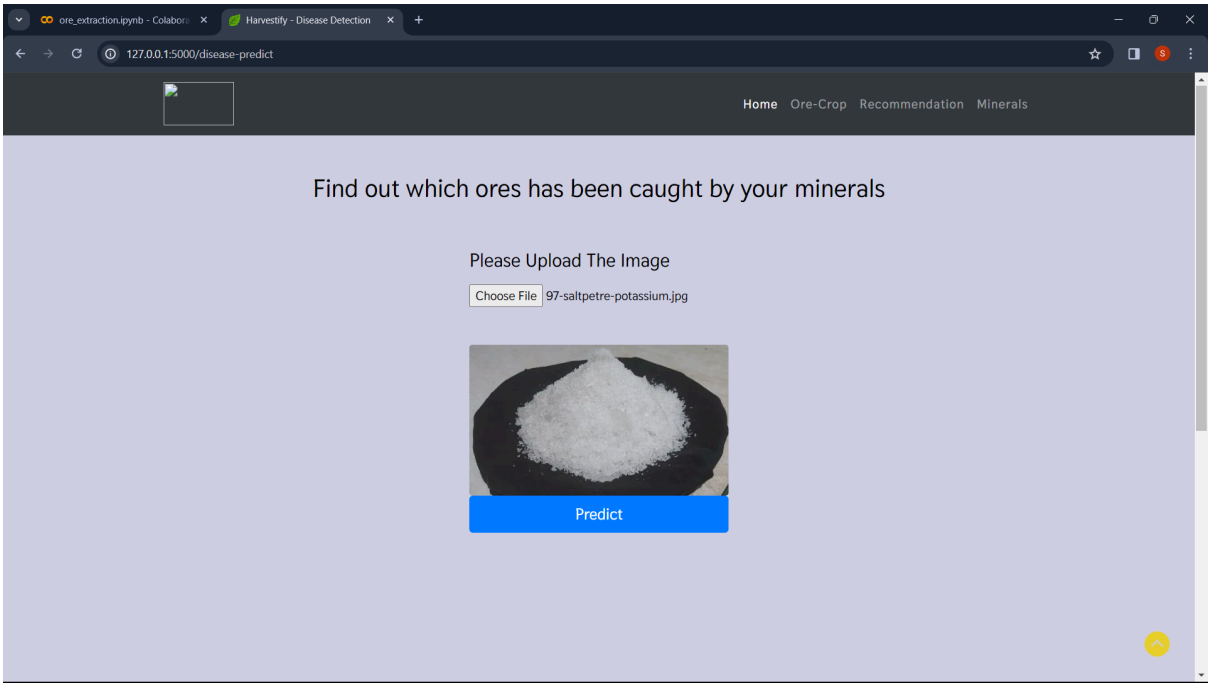


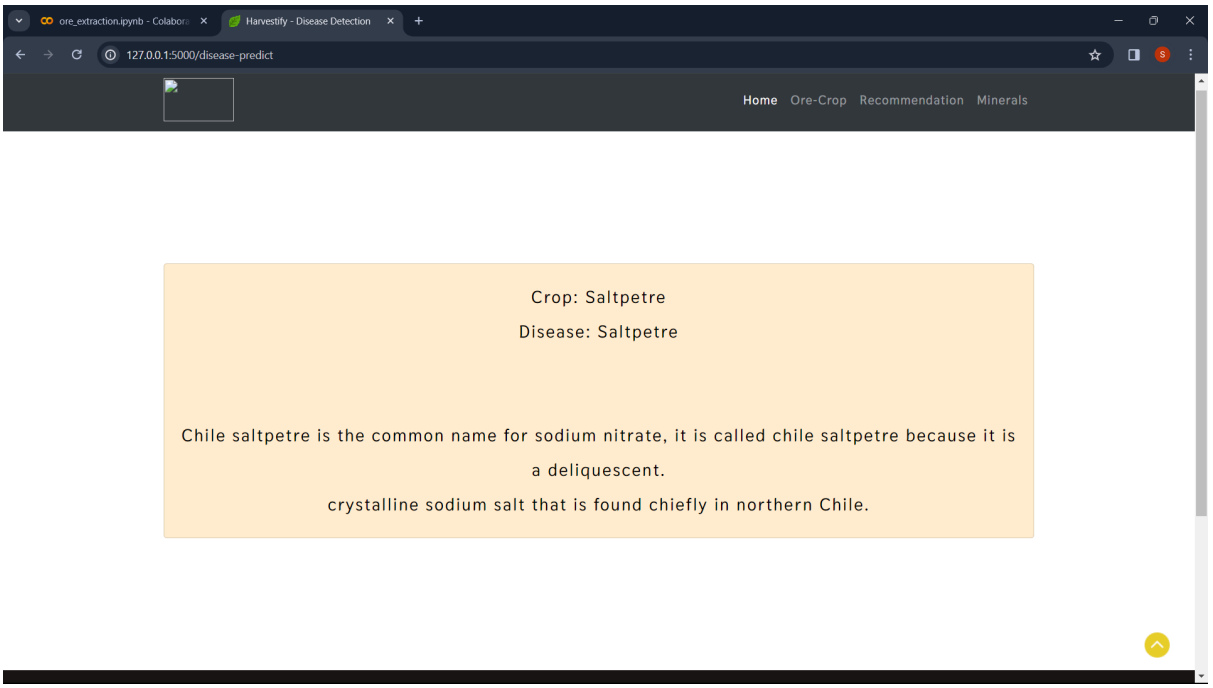



ore_extraction.ipynb - Colabora
Harvestify - Crop Recommenda
127.0.0.1:5000/crop-predict
Home
Ore-Crop
Recommendation
Minerals
You should extract *Chlorargyrite* from your ores
oreclassify
An Environmental Intelligence Startup


ore_extraction.ipynb - Colabora
Harvestify - Fertilizer Suggestio
127.0.0.1:5000/fertilizer
Home
Ore-Crop
Recommendation
Minerals
Get informed advice on ores & Minerals based on soil
Nitrogen
54
Phosphorous
65
Pottasium
22
Ores you want to predict
Rock salt
Predict


ore_extraction.ipynb - Colabora
Harvestify - Fertilizer Suggestio
127.0.0.1:5000/fertilizer-predict
Home
Ore-Crop
Recommendation
Minerals
The K value of your soil is low.
Please consider the following suggestions:
1. Mix in muricate of potash or sulphate of potash
2. Try kelp meal or seaweed
3. Try Sul-Po-Mag
4. Bury banana peels an inch below the soils surface
5. Use Potash fertilizers since they contain high values potassium


ore_extraction.ipynb - Colabora
Harvestify - Disease Detection
127.0.0.1:5000/disease-predict
Home
Ore-Crop
Recommendation
Minerals
Find out which ores has been caught by your minerals
Please Upload The Image
Choose File
97-saltpetre-potassium.jpg
Predict


ore_extraction.ipynb - Colabora
Harvestify - Disease Detection
127.0.0.1:5000/disease-predict
Home
Ore-Crop
Recommendation
Minerals
Crop: Saltpetre
Disease: Saltpetre
Chile saltpetre is the common name for sodium nitrate, it is called chile saltpetre because it is a deliquescent.
crystalline sodium salt that is found chiefly in northern Chile.

PAPER PUBLICATION STATUS

We submitted our paper to ICSSIT-2024 and it got accepted and presented our paper on 6th International Conference on Smart Systems and Inventive Technology ICSSIT 2024

ICSSIT

Certificate of Presentation

This certificate is given to

Mohamed Rizwan

for enthusiastically participating and presenting
a paper titled

Smart Mining System with Crystal Classification of Ores and Industrial Management

in the 6th International Conference on Smart Systems and Inventive Technology (ICSSIT 2024), April 4-5, organized by Francis Xavier Engineering College, Tirunelveli, India.

Session Chair	Conference Chair Dr. G. Rajakumar	Principal Dr. V. Velmurugan

FRANCIS XAVIER®
ENGINEERING COLLEGE
AN AUTONOMOUS INSTITUTION
Vannarpettai, Tirunelveli - 627 003.