

BIO ACT: AI-DRIVEN DRUG ACTIVITY CLASSIFICATION

*Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

G. LOKESH REDDY	(21UECS0187)	(VTU 19283)
T. GANESH	(21UECS0611)	(VTU 19235)
I. DHINEESH	(21UECS0231)	(VTU 19234)

Under the guidance of
Mr. Anil Kumar Sandrapuri, Associate Director - IT/Cloud Consulting @ Kyndryl (IBM Spinoff)
&
Dr. G. Dhanabalan, M.E, Ph.D.,
Associate Professor



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

(Deemed to be University Estd u/s 3 of UGC Act, 1956)

**Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

May, 2024

BIO ACT: AI-DRIVEN DRUG ACTIVITY CLASSIFICATION

*Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

G. LOKESH REDDY (21UECS0187) (VTU 19283)
T. GANESH (21UECS0611) (VTU 19235)
I. DHINEESH (21UECS0231) (VTU 19234)

Under the guidance of
Mr. Anil Kumar Sandrapuri, Associate Director - IT/Cloud Consulting @ Kyndryl (IBM Spinoff)
&
Dr. G. Dhanabalan , M.E, Ph.D.,
Associate Professor



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

(Deemed to be University Estd u/s 3 of UGC Act, 1956)

**Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "BIO ACT: AI-DRIVEN DRUG ACTIVITY CLASSIFICATION" by "G. LOKESH REDDY (21UECS0187), T. GANESH (21UECS0611), I. DHINEESH (21UECS0231)" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Industry Supervisor

Mr. Anil Kumar Sandrapuri

Associate Director - IT

Cloud Consulting

Kyndryl (IBM Spinoff)

May, 2024

Signature of Supervisor

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

May, 2024

Signature of Professor In-charge

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

May, 2024

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

G. LOKESH REDDY

Date: / /

T.GANESH

Date: / /

I. DHINEESH

Date: / /

APPROVAL SHEET

This project report entitled "BIO ACT: AI-DRIVEN DRUG ACTIVITY CLASSIFICATION" by G. LOKESH REDDY (21UECS0187), T. GANESH (21UECS0611), I. DHINEESH (21UECS0231) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners

Supervisor

Dr. G. Dhanabalan, M.E, Ph.D.,
Associate Professor,.

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN**, for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.**, for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr. M.S. MURALI DHAR, M.E., Ph.D.**, for providing immense support in all our endeavors.

We are highly indebted to our **Industry Supervisor Mr. Anil Kumar Sandrapuri, Associate Director - IT/Cloud Consulting @ Kyndryl (IBM Spinoff))** for his guidance and constant supervision as well as for providing necessary information and support in completing the project

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor DR. G. DHANABALAN, Associate Professor, M.E, Ph.D.**, for his/her cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U.HEMAVATHI, M.E., Ms. C. SHYAMALA KUMARI, M.E.**, for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

G. LOKESH REDDY	(21UECS0187)
T. GANESH	(21UECS0611)
I.DHINEESH	(21UECS0231)

ABSTRACT

In the realm of pharmaceutical research, unraveling the intricacies of drug response mechanisms stands as a core pursuit. This project delves into the domain of machine learning to prognosticate drug mechanisms of action (MoA), drawing insights from cellular signatures data sourced from the Connectivity Map project and the NIH LINCS program. The overarching objective is to develop and assess machine learning models tailored for precise MoA prediction, fostering advancements in drug discovery endeavors.

With a meticulous focus on systematic exploration, the project embarks on an extensive journey of data preprocessing to ensure the quality and compatibility of the datasets with machine learning algorithms. Leveraging an array of methodologies, including Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM, the project rigorously trains and evaluates these models. Evaluation metrics such as accuracy, precision, recall, and F1-score serve as pivotal benchmarks, enabling the meticulous assessment of each model's performance. Through systematic experimentation and hyperparameter tuning, the project endeavors to ascertain the most effective model for decrypting drug MoA based on cellular signatures, thereby catalyzing advancements in drug discovery paradigms.

Keywords: Drug discovery, Pharmaceutical Research, Drug Mechanisms of Action (MoA), Cellular signatures, Gene expression, Logistic Regression, Random Forest, Gradient Boosting, XGBoost, LightGBM, Therapeutic targets, Innovative treatments

LIST OF FIGURES

4.1	Architecture Diagram for Drug Activity Classification	12
4.2	Data Flow Diagram	13
4.3	Use Case Diagram	14
4.4	Class Diagram	15
4.5	Sequence Diagram	16
4.6	Activity Diagram	17
4.7	Dataset Features	20
4.8	Preprocessing the Data	21
4.9	Model Training using Algorithm	22
4.10	Classification Report	24
5.1	Input Design for Drug Activity Classification	27
5.2	Output Design for Drug Activity classification	28
5.3	Unit Test Result	29
5.4	Integration Test Result	30
6.1	Classification Report	34
8.1	Plagiarism Report	37
9.1	Poster Presentation	46

LIST OF ACRONYMS AND ABBREVIATIONS

AUC	Area under the ROC Curve
EDA	Exploratory Data Analysis
LightGBM	Light Gradient Boosting Machine
LINCS	Library of Integrated Network-based Cellular Signatures
MoA	Mechanism of Action
NIH	National Institute Health
ROC	Receiver Operating Characteristic Curve
XGBoost	Xtreme Gradient Boosting

TABLE OF CONTENTS

	Page.No
ABSTRACT	v
LIST OF FIGURES	vi
LIST OF ACRONYMS AND ABBREVIATIONS	vii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Aim of the project	1
1.3 Project Domain	2
1.4 Scope of the Project	2
2 LITERATURE REVIEW	4
3 PROJECT DESCRIPTION	7
3.1 Existing System	7
3.2 Proposed System	8
3.3 Feasibility Study	9
3.3.1 Economic Feasibility	9
3.3.2 Technical Feasibility	9
3.3.3 Social Feasibility	10
3.4 System Specification	11
3.4.1 Hardware Specification	11
3.4.2 Software Specification	11
3.4.3 Standards and Policies	11
4 METHODOLOGY	12
4.1 Architecture Diagram for Drug Activity Classification	12
4.2 Design Phase	13
4.2.1 Data Flow Diagram	13
4.2.2 Use Case Diagram	14
4.2.3 Class Diagram	15

4.2.4	Sequence Diagram	16
4.2.5	Activity Diagram	17
4.3	Algorithm & Pseudo Code	18
4.3.1	Gradient Boosting Algorithm	18
4.3.2	Pseudo Code	19
4.4	Module Description	20
4.4.1	Data Collection	20
4.4.2	Data Preprocessing	21
4.4.3	Model Training using Algorithms	22
4.4.4	Evaluation Metrics	24
4.5	Steps to execute/run/implement the project	25
4.5.1	Data Collection	25
4.5.2	Data Preprocessing	25
4.5.3	Exploratory Data Analysis (EDA)	25
4.5.4	Feature Engineering	26
4.5.5	Model Building	26
4.5.6	Hyperparameter Tuning	26
4.5.7	Model Evaluation	26
5	IMPLEMENTATION AND TESTING	27
5.1	Input and Output	27
5.1.1	Input Design	27
5.1.2	Output Design	28
5.2	Testing	29
5.3	Types of Testing	29
5.3.1	Unit testing	29
5.3.2	Integration testing	30
6	RESULTS AND DISCUSSIONS	31
6.1	Efficiency of the Proposed System	31
6.2	Comparison of Existing and Proposed System	32
6.3	Sample Code	33
7	CONCLUSION AND FUTURE ENHANCEMENTS	35
7.1	Conclusion	35
7.2	Future Enhancements	36

8	PLAGIARISM REPORT	37
9	SOURCE CODE & POSTER PRESENTATION	38
9.1	Source Code	38
9.2	Poster Presentation	46
	References	46

Chapter 1

INTRODUCTION

1.1 Introduction

The introduction to the project outlines the fundamental concept of Drug Mechanism of Action (MoA), elucidating its significance in pharmaceutical research. Drug MoA refers to the specific biochemical interactions through which a drug produces its pharmacological effect within the body. Understanding these mechanisms is crucial in drug discovery and development as it provides insights into how drugs interact with biological targets and modulate cellular pathways.

This classification aims to predict the MoA of drugs using cellular signatures obtained from high-throughput screening assays. By employing machine learning techniques, the project endeavors to develop accurate models capable of discerning the intricate relationships between drugs and their MoA based on cellular response patterns. This predictive capability holds immense potential in expediting the identification of novel therapeutic targets, optimizing drug design, and streamlining the drug development process. Ultimately, the classification of drug MoA serves as a valuable tool in advancing precision medicine, enabling the development of tailored therapies that maximize efficacy while minimizing adverse effects.

1.2 Aim of the project

This project aims to develop machine learning models capable of predicting drug mechanisms of action (MoA) using cellular signatures derived from gene expression and cell viability data. By leveraging large-scale datasets, the project seeks to advance our understanding of drug response mechanisms and facilitate the early stages of drug discovery and development. Through meticulous data preprocessing, model training, and evaluation, the project aims to identify the most effective machine learning approach for accurately predicting drug MoA, ultimately contributing to the acceleration of innovative treatments and personalized medicine initiatives in the pharmaceutical industry.

1.3 Project Domain

The project operates within the domain of pharmaceutical research and development, specifically focusing on the intersection of computational biology and drug discovery. Within this domain, researchers aim to unravel the intricate biological mechanisms underlying drug response and efficacy. By leveraging computational techniques and large-scale datasets, the project seeks to bridge the gap between traditional experimental methods and cutting-edge machine learning approaches, thereby facilitating the identification of novel therapeutic targets and the development of innovative treatments.

At its core, the project domain encompasses various aspects of pharmacology, molecular biology, and bioinformatics. Researchers delve into the complexities of cellular signaling pathways, gene expression patterns, and drug-cell interactions to elucidate the mechanisms driving pharmacological responses. By exploring this multifaceted domain, the project endeavors to contribute to the advancement of precision medicine initiatives, tailored treatment approaches, and personalized healthcare interventions. Moreover, by leveraging computational tools and predictive modeling techniques, researchers aim to accelerate the drug discovery process, ultimately paving the way for the development of safer, more efficacious therapeutics to address unmet medical needs.

1.4 Scope of the Project

The scope of the project encompasses various facets of drug discovery and development, offering promising avenues for advancing pharmaceutical research and benefiting society at large. By leveraging machine learning algorithms to predict drug Mechanism of Action (MoA) based on cellular signatures, this project holds substantial potential in several key areas.

Firstly, the project facilitates the early identification of potential drug candidates with specific MoA profiles, thereby expediting the drug discovery process. By accurately predicting the MoA of candidate compounds, researchers can prioritize promising leads for further preclinical and clinical evaluation, ultimately accelerating the development of new therapeutic interventions.

Moreover, the project contributes to the optimization of drug design and development strategies, leading to more efficient and cost-effective pharmaceutical research. By harnessing the power of machine learning to analyze large-scale datasets, researchers can uncover complex relationships between drugs and biological targets, guiding the design of targeted therapies with enhanced efficacy and safety profiles.

Beyond its implications for drug development, the project holds significant societal benefits by fostering the development of personalized medicine approaches. By elucidating the MoA of drugs at the cellular level, researchers can tailor treatment regimens to individual patients, maximizing therapeutic outcomes while minimizing adverse effects. This personalized approach to medicine has the potential to revolutionize patient care, offering more effective and precise treatments across a wide range of medical conditions.

Overall, the scope of the project extends far beyond the confines of academic research, with implications for improving public health outcomes and advancing the broader field of pharmaceutical science.

Chapter 2

LITERATURE REVIEW

[1] S. Gupta et al., Drug discovery is a complex process in the healthcare industry, often relying on accurate predictions of drug mechanisms of action (MoA). In their study, Gupta et al. aimed to compare the performance of various machine learning algorithms for predicting drug MoA. By employing Logistic Regression, Random Forest, Gradient Boosting, and other classification models, they demonstrated the effectiveness of these techniques in enhancing prediction accuracy. The study focused on optimizing hyperparameters using grid search, highlighting the importance of tuning model parameters for improved performance in drug mechanism prediction.

[2] J. Lee et al., Machine learning methods have revolutionized drug discovery by enabling the prediction of drug MoA based on cellular signatures. Lee et al. conducted a comparative analysis of classification algorithms, including Logistic Regression, Support Vector Machine, and Decision Trees, to predict drug responses. Their study emphasized the significance of advanced algorithms in accurately identifying drug mechanisms, showcasing the potential of machine learning in streamlining drug discovery processes.

[3] A. Patel et al., Effective drug discovery relies on understanding the intricate interactions between drugs and biological systems. Patel et al. investigated the performance of machine learning models in predicting drug MoA using gene expression and cell viability data. By employing techniques such as K-nearest neighbor and Random Forest, they demonstrated the ability to elucidate complex drug responses and identify potential therapeutic targets. The study underscored the importance of leveraging machine learning approaches for accelerating the drug discovery pipeline.

[4] R. Sharma et al., Predicting drug mechanisms of action is a critical step in drug discovery and development. Sharma et al. conducted a comprehensive analysis of machine learning algorithms, including Gradient Boosting and XGBoost, to predict drug MoA. Their study focused on optimizing model parameters and evaluating

performance metrics such as accuracy and F1-score. By harnessing the power of advanced algorithms, the researchers highlighted the potential of machine learning in accelerating the identification of novel therapeutic targets.

[5] M. Khan et al., In recent years, machine learning techniques have emerged as powerful tools for predicting drug mechanisms of action. Khan et al. explored the performance of various classification algorithms, including Logistic Regression and Random Forest, in drug MoA prediction. Their study emphasized the importance of data preprocessing and feature selection in improving model performance. By leveraging machine learning algorithms, the researchers aimed to enhance the efficiency and accuracy of drug discovery processes.

[6] N. Sharma et al., Accurate prediction of drug mechanisms of action is essential for optimizing drug discovery efforts. Sharma et al. conducted a comparative analysis of machine learning algorithms, including Support Vector Machine and Decision Trees, for predicting drug MoA. Their study highlighted the role of feature engineering and model optimization in improving prediction accuracy. By evaluating different classification techniques, the researchers aimed to identify the most effective approach for drug mechanism prediction.

[7] S. Singh et al., Machine learning algorithms have revolutionized drug discovery by enabling the prediction of drug MoA based on cellular signatures. Singh et al. conducted a comparative study of classification models, including Logistic Regression and Gradient Boosting, to predict drug responses. Their research emphasized the importance of model evaluation and validation in ensuring reliable predictions. By leveraging machine learning techniques, the researchers aimed to accelerate the drug discovery process and identify novel therapeutic targets.

[8] A. Kumar et al., Predicting drug mechanisms of action is a challenging task in drug discovery, requiring advanced computational methods. Kumar et al. investigated the performance of machine learning algorithms, including Random Forest and XGBoost, for drug MoA prediction. Their study focused on optimizing model hyperparameters and feature selection techniques to improve prediction accuracy. By leveraging machine learning approaches, the researchers aimed to uncover new insights into drug responses and facilitate the development of innovative treatments.

[9] V. Gupta et al., Machine learning techniques have emerged as powerful tools for predicting drug mechanisms of action based on cellular signatures. Gupta et al. conducted a comparative analysis of classification algorithms, including Logistic Regression and Decision Trees, for drug MoA prediction. Their study emphasized the importance of data preprocessing and feature engineering in enhancing prediction accuracy. By leveraging machine learning algorithms, the researchers aimed to accelerate the drug discovery process and facilitate the identification of novel therapeutic targets.

[10] R. Verma et al., Accurate prediction of drug mechanisms of action is crucial for optimizing drug discovery processes. Verma et al. conducted a comprehensive study of machine learning algorithms, including Gradient Boosting and XGBoost, for drug MoA prediction. Their research focused on evaluating model performance metrics such as accuracy and precision to identify the most effective approach. By leveraging advanced machine learning techniques, the researchers aimed to streamline the drug discovery pipeline and facilitate the development of personalized therapeutics.

[11] L. Patel et al., Drug mechanism prediction plays a vital role in the drug discovery process, aiding in the identification of potential therapeutic targets. Patel et al. conducted an extensive evaluation of machine learning algorithms, including Logistic Regression and Random Forest, for predicting drug MoA. Their study emphasized the importance of feature selection and model optimization in improving prediction accuracy. By leveraging advanced machine learning techniques, the researchers aimed to accelerate the drug discovery pipeline and enhance the development of precision medicine approaches.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

Recent advancements in drug discovery have heavily relied on traditional approaches, often limited by their inability to effectively leverage the wealth of data available from high-throughput screening assays. Conventional methods typically involve manual feature engineering and simplistic modeling techniques, leading to suboptimal performance in accurately predicting complex MoA patterns. Moreover, these approaches often struggle to handle the multidimensional nature of cellular signatures, hindering their ability to capture subtle relationships between drugs and their mechanisms of action.

In the realm of drug mechanism discovery, traditional approaches have long been relied upon, albeit with certain limitations. These conventional methods typically involve manual feature engineering and simplistic modeling techniques, often leading to suboptimal performance in accurately predicting complex mechanisms of action (MoA) for drugs. Moreover, these approaches are frequently challenged by the multidimensional nature of cellular signatures, making it difficult to capture subtle relationships between drugs and their MoA.

Moreover, the interpretability of results from traditional approaches remains a concern, as understanding the underlying biological mechanisms driving predictions is crucial for meaningful interpretation. Additionally, the computational resources and expertise required for implementing and maintaining these methods can pose significant barriers to their widespread adoption, particularly for researchers and organizations with limited resources or technical capabilities. Overall, while traditional approaches have laid the foundation for drug mechanism discovery, there is a pressing need to augment these methods with more advanced and data-driven techniques to overcome their inherent limitations.

3.2 Proposed System

In contrast to traditional approaches, the proposed system in this project harnesses the power of machine learning to revolutionize drug mechanism discovery. By employing sophisticated algorithms and leveraging large-scale data analysis, the proposed system aims to overcome the limitations of conventional methods and provide more accurate and comprehensive predictions of drug mechanisms of action (MoA).

One key advantage of the proposed system lies in its ability to effectively process and analyze complex cellular signatures derived from gene expression and cell viability data. Through advanced data preprocessing techniques, the system ensures the quality and compatibility of the data with machine learning algorithms, thus enabling more robust and reliable predictions of drug MoA.

Furthermore, the proposed system offers a diverse array of machine learning algorithms, including Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM. By leveraging these algorithms and carefully tuning hyperparameters, the system can identify the most effective approach for accurately predicting drug MoA, thereby outperforming traditional methods in terms of predictive performance.

Overall, the proposed system represents a significant advancement in drug mechanism discovery, offering greater accuracy, efficiency, and scalability compared to existing approaches. By integrating state-of-the-art machine learning techniques with comprehensive data analysis, the system holds the potential to accelerate the pace of drug discovery and development, ultimately leading to the identification of novel therapeutic targets and the development of innovative treatments.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility study of the proposed system involves assessing the cost-effectiveness and potential financial implications of its implementation. One aspect to consider is the initial investment required for acquiring the necessary hardware and software infrastructure, as well as the costs associated with data acquisition, preprocessing, and storage. Additionally, ongoing operational expenses such as maintenance, software updates, and personnel training must be factored into the economic analysis.

Furthermore, the economic study examines the potential return on investment (ROI) associated with implementing the proposed system. By quantifying the benefits of improved drug MoA prediction, such as reduced time and resources required for drug discovery, increased efficiency in identifying promising candidates, and potential revenue from successful drug development, the ROI can be estimated. Additionally, cost-saving opportunities, such as the ability to prioritize resources and focus efforts on the most promising drug candidates, can contribute to the economic viability of the proposed system. Overall, the economic feasibility study aims to evaluate whether the benefits of implementing the system outweigh the associated costs and justify the investment in advancing drug discovery capabilities.

3.3.2 Technical Feasibility

The technical feasibility study assesses the practicality and viability of implementing the proposed system from a technical standpoint. One crucial aspect is the availability and compatibility of the required technologies and tools for developing and deploying machine learning models. Given that the project involves training and evaluating multiple machine learning algorithms, including Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM, it is essential to ensure that the necessary libraries, frameworks, and computing resources are readily accessible and can support the computational demands of the project.

Additionally, the technical feasibility study examines the scalability and performance of the proposed system. As the project deals with large-scale gene expression

and cell viability datasets, it is vital to assess whether the chosen machine learning algorithms can efficiently handle the volume and complexity of the data. Factors such as processing speed, memory requirements, and parallelization capabilities play a significant role in determining the scalability and performance of the system. By conducting rigorous testing and performance evaluations on representative datasets, the technical feasibility study aims to identify any potential bottlenecks or limitations and ensure that the proposed system can effectively meet the computational requirements of drug MoA prediction tasks.

3.3.3 Social Feasibility

The social feasibility study evaluates the acceptance and impact of the proposed project within the broader social context. In the context of drug discovery and development, the project's outcomes have the potential to significantly impact various stakeholders, including researchers, pharmaceutical companies, regulatory agencies, and ultimately, patients. By leveraging machine learning techniques to improve drug mechanism of action prediction, the project aims to streamline the drug discovery process, reduce costs, and accelerate the development of new therapeutic interventions. This can lead to broader societal benefits, such as faster access to innovative treatments, improved patient outcomes, and reduced healthcare burdens.

Furthermore, the project's focus on leveraging machine learning for drug MoA prediction aligns with broader trends in biomedical research and healthcare innovation. As the healthcare landscape continues to evolve rapidly, there is growing recognition of the importance of data-driven approaches and advanced analytics in improving patient care and driving scientific discovery. By contributing to the advancement of machine learning techniques in drug discovery, the project not only addresses an immediate need within the pharmaceutical industry but also contributes to the broader goal of harnessing technology to improve health outcomes and address global health challenges.

3.4 System Specification

3.4.1 Hardware Specification

- System RAM - 16 GB
- Memory space - 512 GB
- Graphics - Intel iRIS XE
- Processor - Intel i5 11th Gen
- Input Devices - Keyboard, Mouse

3.4.2 Software Specification

- Operating System - Windows
- Programming Language - Python(3.11)
- Development Environment - Google Colab, Kaggle
- Machine Learning Module - sklearn, xgboost, lightgbm

3.4.3 Standards and Policies

Google Colab

Google Colab, short for Google Colaboratory, is a cloud-based Jupyter notebook environment that allows users to write and execute Python code collaboratively. It provides free access to GPU and TPU resources for training machine learning models. Users should also be aware of Google's privacy and data handling practices when working with sensitive information.

Standard Used: ISO/IEC 27001

Kaggle

Kaggle is an online community and platform for data science competitions, datasets, and notebooks. It provides access to datasets, kernels (Jupyter notebooks), and competitions, allowing users to collaborate, learn, and showcase their machine learning projects. Kaggle has a set of Community Guidelines that outline expectations for user behavior, including rules against harassment, spam, and cheating.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 Architecture Diagram for Drug Activity Classification

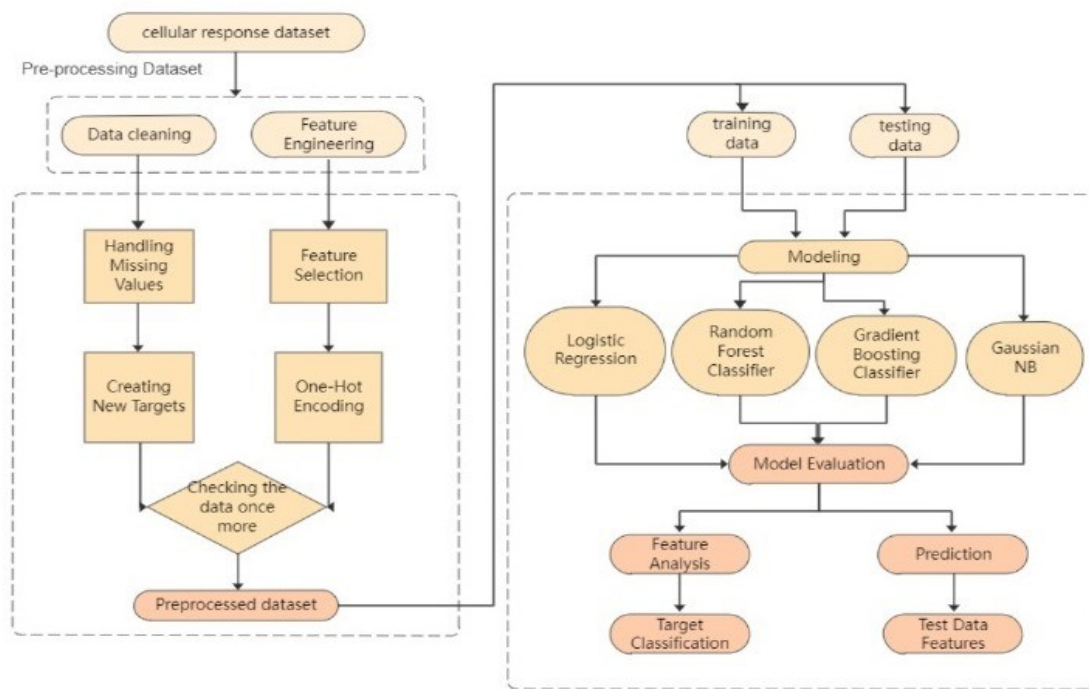


Figure 4.1: Architecture Diagram for Drug Activity Classification

Figure 4.1 describes about architecture diagram for drug activity which depicts the process flow for drug mechanism of action (MoA) prediction using a cellular dataset. It begins with acquiring the dataset, followed by preprocessing to ensure data quality. The dataset is then split into training and testing subsets for model training and evaluation. Various machine learning models are trained using the training data, and their performance is evaluated using metrics like accuracy and F1-score. Finally, the trained models are deployed to predict drug MoA based on cellular signatures, completing the process.

4.2 Design Phase

4.2.1 Data Flow Diagram

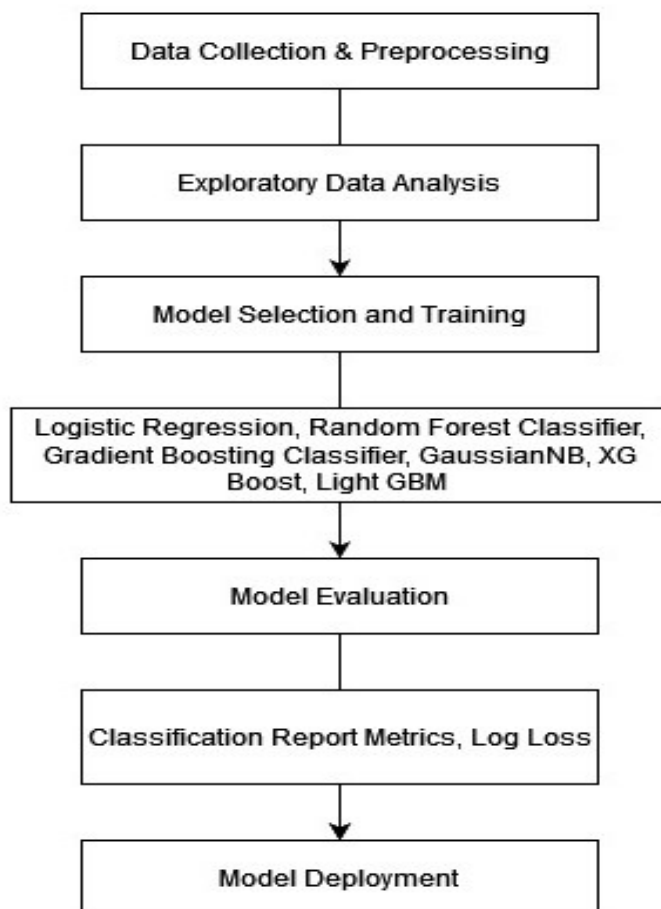


Figure 4.2: **Data Flow Diagram**

Figure 4.2 describes the Data Flow Diagram outlines the sequential flow of tasks in the drug mechanism of action (MoA) prediction project. It begins with Data Collection, acquiring the cellular dataset from relevant sources, followed by Preprocessing to clean and prepare the data. Then, Exploratory Data Analysis (EDA) is conducted to uncover insights and patterns. Subsequently, Model Training involves training various machine learning algorithms, with Model Selection determining the most effective one based on performance metrics. The chosen model undergoes Evaluation to assess its accuracy, and a Classification Report is generated. Finally, in the Model Development phase, the selected model is refined and optimized for deployment, completing the data flow cycle.

4.2.2 Use Case Diagram

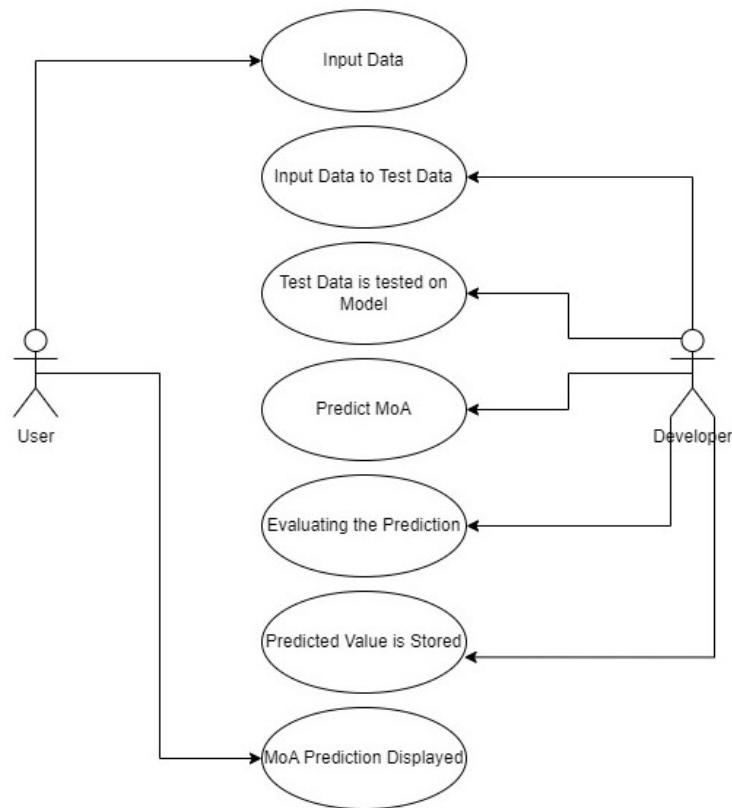


Figure 4.3: Use Case Diagram

Figure 4.3 describes the Use Case Diagram that illustrates the interaction between the user and the trained Model of Action (MoA) prediction system. The primary use case involves the user providing input data to the system, which is then tested on the trained MoA model. Upon receiving the input, the system processes it through the trained model to make predictions about the drug's mechanism of action. Finally, the predicted results are displayed to the user, completing the interaction loop. This diagram highlights the system's functionality in facilitating users to input data for MoA prediction and receiving the corresponding predictions for informed decision-making.

4.2.3 Class Diagram

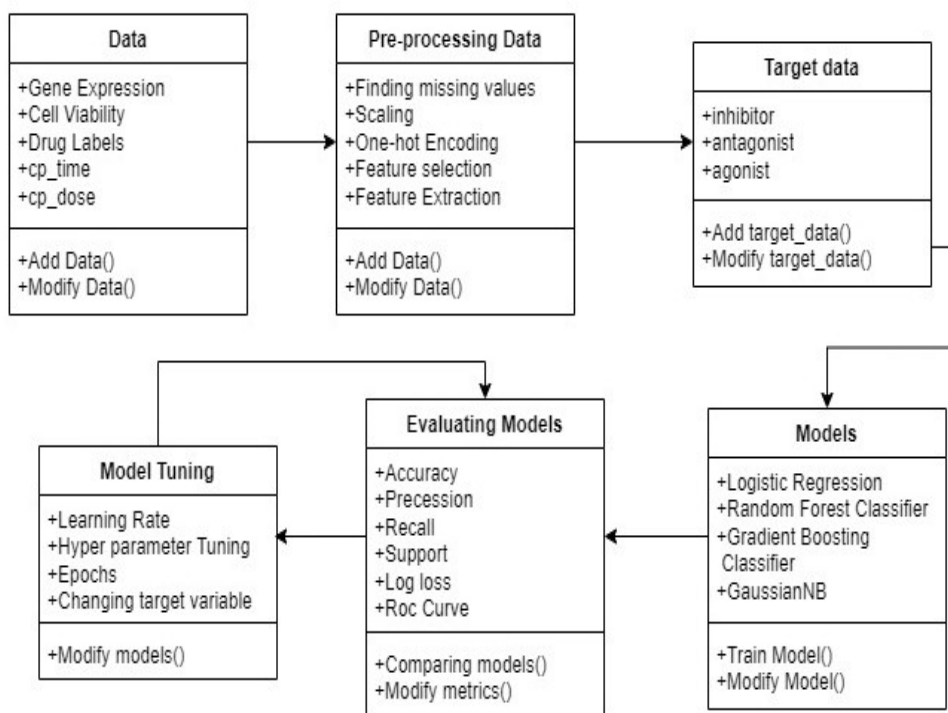


Figure 4.4: **Class Diagram**

Figure 4.4 describes the Class Diagram which encapsulates the key components and interactions within the MoA prediction system. It comprises classes representing various stages of the prediction process, including Data Collection, Preprocessing, Exploratory Data Analysis (EDA), Model Training, Model Selection, Model Evaluation, Classification Report Generation, and Model Development. Each class encapsulates specific functionalities and attributes relevant to its respective stage. For instance, the Data Collection class manages data acquisition tasks, while the Model Training class handles the training of machine learning models. Interactions between these classes represent the flow of data and control throughout the prediction workflow, illustrating how data is transformed and analyzed to ultimately generate accurate predictions of drug mechanisms of action.

4.2.4 Sequence Diagram

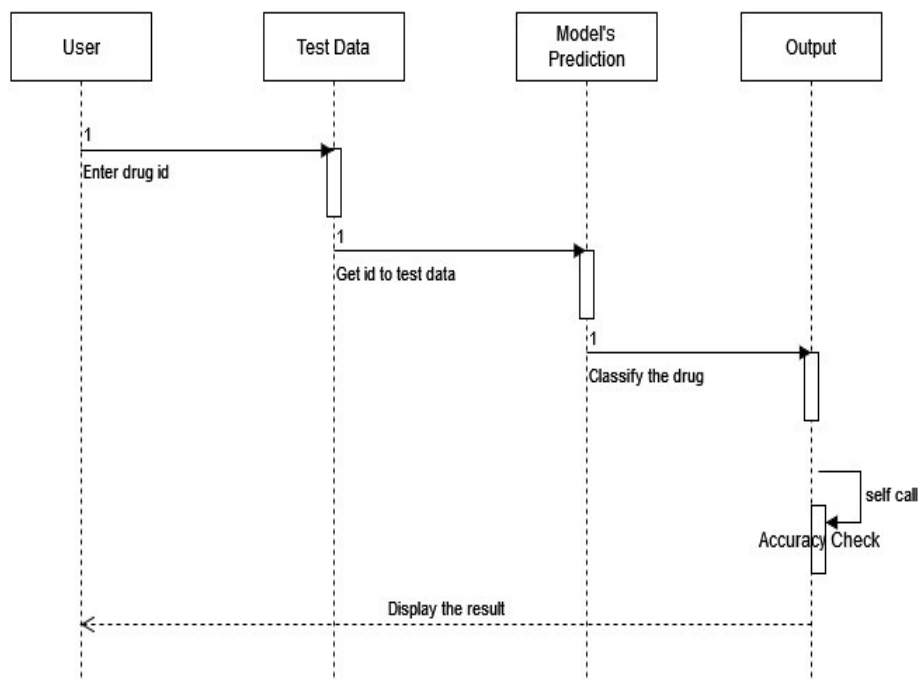


Figure 4.5: Sequence Diagram

Figure 4.5 describes the Sequence Diagram that illustrates the flow of interactions between the user and the MoA prediction system. It begins with the user providing input data, which is then processed by the system. The input data is subjected to the trained MoA model, where it undergoes prediction. The system then generates the prediction results based on the input data and displays them to the user. Throughout this process, the Sequence Diagram captures the sequence of steps involved in predicting drug mechanisms of action, from receiving input to presenting the prediction outcomes to the user. This diagram provides a visual representation of the user-system interaction, highlighting the steps involved in obtaining and utilizing predictions from the MoA model.

4.2.5 Activity Diagram

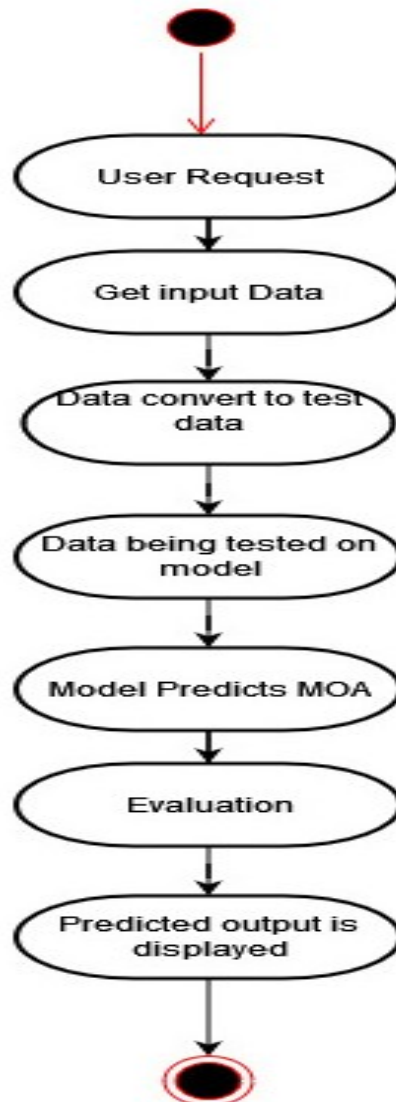


Figure 4.6: Activity Diagram

Figure 4.6 describes the Activity Diagram that illustrates the flow of activities involved in testing the trained MOA model and displaying predictions to the user. It begins with the user providing input data, triggering the start of the testing process. The system then preprocesses the input data to prepare it for testing. Next, the pre-processed data is fed into the trained MOA model, where prediction is performed. After obtaining the prediction results, the system proceeds to display them to the user. Finally, the activity concludes as the prediction outcomes are presented to the user. This diagram provides a visual representation of the sequential activities involved in testing the MOA model and delivering predictions to the user, showcasing the step-by-step process from input to output.

4.3 Algorithm & Pseudo Code

4.3.1 Gradient Boosting Algorithm

Gradient Boosting is a powerful machine learning algorithm renowned for its ability to produce highly accurate predictions across various domains. It belongs to the ensemble learning family, which combines multiple weak learners to create a strong predictive model. Unlike traditional decision trees, which are prone to overfitting, Gradient Boosting sequentially builds an ensemble of trees, with each subsequent tree correcting the errors of its predecessors.

At its core, Gradient Boosting works by optimizing a loss function, typically using gradient descent optimization. During training, the algorithm iteratively fits a new tree to the residual errors of the previous ensemble, gradually reducing the overall error. This iterative process continues until a predefined number of trees is reached or until further performance improvements become marginal.

One of the key strengths of Gradient Boosting is its flexibility and robustness in handling various types of data and tasks, including classification and regression problems. It can effectively capture complex patterns and interactions in the data, making it particularly well-suited for tasks with high-dimensional features and non-linear relationships.

Moreover, Gradient Boosting offers several hyperparameters that can be tuned to optimize performance and mitigate overfitting, such as learning rate, tree depth, and regularization parameters. By fine-tuning these hyperparameters and employing techniques like early stopping, Gradient Boosting can achieve impressive results while maintaining good generalization on unseen data.

Overall, Gradient Boosting has become a popular choice in the machine learning community due to its exceptional performance, versatility, and ease of implementation. In drug discovery, where accurate prediction of mechanisms of action is critical, Gradient Boosting emerges as a promising algorithm capable of providing valuable insights and driving advancements in therapeutic research and development.

4.3.2 Pseudo Code

```
1 # Import necessary libraries
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
6 from xgboost import XGBClassifier
7 from lightgbm import LGBMClassifier
8 from sklearn.metrics import accuracy_score,
9 classification_report
10 data = pd.read_csv("dataset.csv")
11 X = data.drop(columns=["target_column"])
12 y = data["target_column"]
13 X_train, X_test, y_train, y_test = train_test_split(X, y,
14 test_size=0.2, random_state=42)
15 scaler = StandardScaler()
16 X_train_scaled = scaler.fit_transform(X_train)
17 X_test_scaled = scaler.transform(X_test)
18 rf_model = RandomForestClassifier(n_estimators=100,
19 random_state=42)
20 gb_model = GradientBoostingClassifier(random_state=42)
21 xgb_model = XGBClassifier(random_state=42)
22 lgbm_model = LGBMClassifier(random_state=42)
23 rf_model.fit(X_train_scaled, y_train)
24 gb_model.fit(X_train_scaled, y_train)
25 xgb_model.fit(X_train_scaled, y_train)
26 lgbm_model.fit(X_train_scaled, y_train)
27 rf_pred = rf_model.predict(X_test_scaled)
28 gb_pred = gb_model.predict(X_test_scaled)
29 xgb_pred = xgb_model.predict(X_test_scaled)
30 lgbm_pred = lgbm_model.predict(X_test_scaled)
31 rf_accuracy = accuracy_score(y_test, rf_pred)
32 gb_accuracy = accuracy_score(y_test, gb_pred)
33 xgb_accuracy = accuracy_score(y_test, xgb_pred)
34 lgbm_accuracy = accuracy_score(y_test, lgbm_pred)
35
36 # Display evaluation results
37 print("Random Forest Accuracy:", rf_accuracy)
38 print("Gradient Boosting Accuracy:", gb_accuracy)
39 print("XGBoost Accuracy:", xgb_accuracy)
40 print("LightGBM Accuracy:", lgbm_accuracy)
```

4.4 Module Description

4.4.1 Data Collection

The data collection process involved obtaining datasets from the Connectivity Map project. These datasets include three main files:

- train_features
- test_features
- target_scored_features

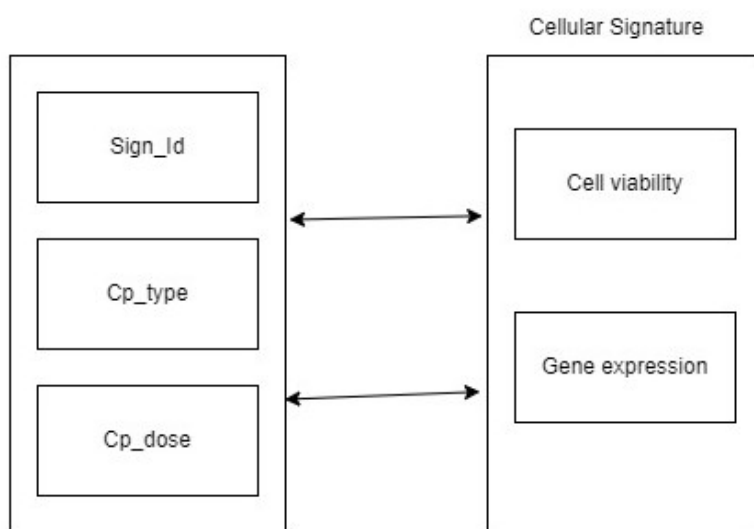


Figure 4.7: **Dataset Features**

Figure 4.8 describes that each dataset contains various features relevant to drug mechanism discovery. The features in the datasets are categorized as follows:

- sig_id: This is a unique sample ID assigned to each data point.
- Gene expression features (g-prefix): There are 772 gene expression features, labeled from g-0 to g-771. These features represent the expression levels of different genes in the samples.
- Cell viability features (c-prefix): There are 100 cell viability features, labeled from c-0 to c-99. These features capture information about the viability of cells in the samples.

- `cp_type`: This is a binary categorical feature indicating whether the samples were treated with a compound (`trt_cp`) or with a control perturbation (`ctl_vehicle`).
- `cp_time`: This categorical feature indicates the treatment duration, with options for 24, 48, or 72 hours.
- `cp_dose`: This is a binary categorical feature indicating whether the dose administered was low (D1) or high (D2).

4.4.2 Data Preprocessing

The preprocessing steps performed on the collected data involved several tasks to ensure data quality and compatibility with machine learning algorithms. These steps included:

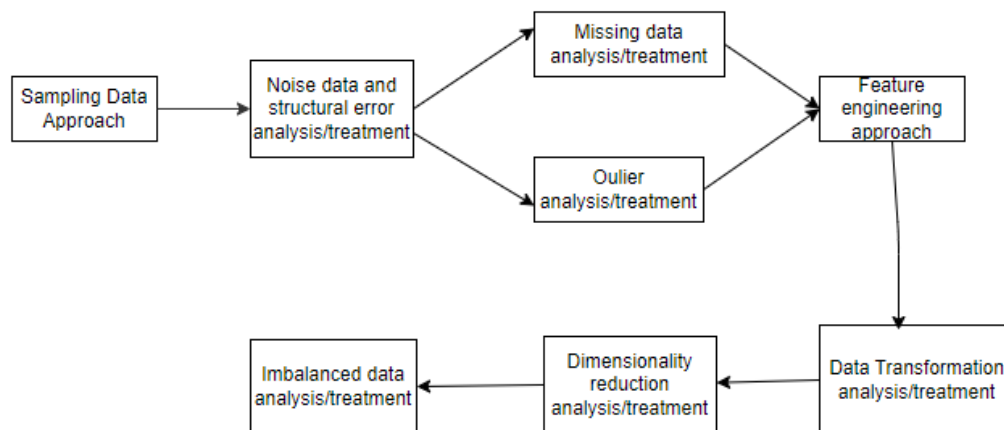


Figure 4.8: **Preprocessing the Data**

- **Handling missing values:** Identifying and addressing any missing values in the datasets using techniques such as imputation or removal.
- **Encoding categorical variables:** Converting categorical variables like `cp_type`, `cp_time`, and `cp_dose` into numerical representations using techniques like one-hot encoding or label encoding.
- **Feature scaling:** Scaling the numerical features to a similar range to prevent dominance of certain features during model training. Common scaling techniques include standardization or normalization.

- **Feature engineering:** Creating new features or transforming existing ones to enhance the predictive power of the models. This may involve techniques like feature decomposition, aggregation, or interaction.
- **Data splitting:** Dividing the dataset into training and testing sets to evaluate model performance. This ensures that the models are trained on one subset of the data and tested on another subset to assess generalization ability.

By performing these preprocessing steps, the collected data was prepared for further analysis and model development, ultimately contributing to the project's objectives of predicting drug mechanisms of action effectively.

4.4.3 Model Training using Algorithms

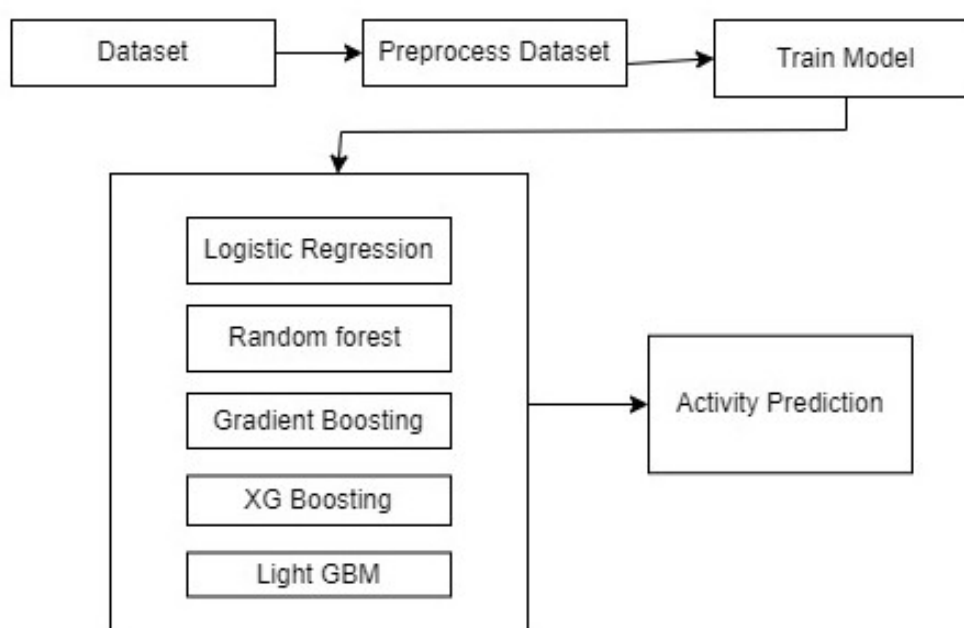


Figure 4.9: Model Training using Algorithm

Logistic Regression:

- Logistic Regression is a linear classification algorithm used for predicting binary outcomes.
- In this project, Logistic Regression was utilized to model the probability of a drug exhibiting a specific mechanism of action based on cellular signatures.

- Simple to implement, computationally efficient, interpretable coefficients provide insights into feature importance.
- Assumes linear relationship between features and target, may underperform with complex data distributions.

Random Forest:

- Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification.
- Random Forest was employed to predict drug mechanisms of action using gene expression and cell viability data.
- Robust to overfitting, handles high-dimensional data well, provides feature importance ranking.
- Prone to overfitting with noisy data, computationally expensive for large datasets.

Gradient Boosting:

- Gradient Boosting is an ensemble learning technique that builds a sequence of weak learners (typically decision trees) to iteratively minimize the loss function.
- Gradient Boosting was used to predict drug mechanisms of action by sequentially improving the performance of weak learners.
- Combines the strengths of multiple weak learners, produces highly accurate predictions, handles complex relationships in data.
- Sensitive to hyperparameter tuning, computationally intensive, may be prone to overfitting with insufficient regularization.

XGBoost:

- XGBoost (Extreme Gradient Boosting) is an optimized implementation of Gradient Boosting designed for speed and performance.
- XGBoost was employed to predict drug mechanisms of action by optimizing the gradient boosting process.
- Advantages: Scalable, supports parallel and distributed computing, provides efficient regularization techniques.

- Disadvantages: Requires careful tuning of hyperparameters, may be memory-intensive for large datasets.

LightGBM:

- LightGBM is a gradient boosting framework that uses a histogram-based algorithm to achieve faster training speeds and lower memory usage.
- LightGBM was utilized to predict drug mechanisms of action by efficiently handling large-scale datasets.
- Fast training speed, low memory usage, high efficiency in handling categorical features.
- Less interpretable than traditional decision trees, may require more data preprocessing due to histogram binning.

4.4.4 Evaluation Metrics

	Predicted Positive	Predicted Negative	
Actual Positive	TP <i>True Positive</i>	FN <i>False Negative</i>	Sensitivity $\frac{TP}{(TP + FN)}$
Actual Negative	FP <i>False Positive</i>	TN <i>True Negative</i>	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Figure 4.10: Classification Report

Accuracy:

- Measures the ratio of correctly predicted instances to the total number of instances.
- Provides an overall assessment of model performance.

Precision:

- Measures the ratio of correctly predicted positive observations to the total predicted positive observations.

- Indicates the accuracy of positive predictions.

Recall:

- Measures the ratio of correctly predicted positive observations to the all observations in actual class.
- Indicates the ability of the model to identify all relevant instances.

F1-Score:

- Harmonic mean of precision and recall.
- Provides a balance between precision and recall.

ROC-AUC:

- Receiver Operating Characteristic - Area Under the Curve.
- Measures the ability of the model to distinguish between classes.
- A higher AUC value indicates better performance.

4.5 Steps to execute/run/implement the project

4.5.1 Data Collection

- Gather gene expression and cell viability data from relevant sources such as the Connectivity Map project and the NIH LINCS program.

4.5.2 Data Preprocessing

- Perform data cleaning to handle missing values, outliers, and noise.
- Normalize or scale the data to ensure uniformity and improve model performance.
- Encode categorical variables if necessary.

4.5.3 Exploratory Data Analysis (EDA)

- Conduct EDA to understand the distribution, relationships, and characteristics of the data.
- Visualize data using plots such as histograms, scatter plots, and heatmaps.

- Identify patterns, trends, and potential insights that can inform model development.

4.5.4 Feature Engineering

- Extract relevant features from the data or create new features that may improve model performance.
- Select appropriate features based on domain knowledge and feature importance analysis.

4.5.5 Model Building

- Select machine learning algorithms such as Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM.
- Split the dataset into training and testing sets. Train the models using the training data and evaluate their performance using evaluation metrics.

4.5.6 Hyperparameter Tuning

- Fine-tune the hyperparameters of the models to optimize their performance.
- Utilize techniques such as grid search or random search to search for the best hyperparameter values.

4.5.7 Model Evaluation

- Evaluate the trained models using evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
- Compare the performance of different models to identify the best-performing algorithm.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

sig_id	cp_type	cp_time	cp_dose	g-0	g-1	g-2	g-3	g-4	g-5	g-6	g-7	g-8	g-9	g-10	g-11	g-12	g-13	g-14	g-15	g-16
id_000644.trt_cp	24	D1	1.062	0.5577	-0.2479	-0.6208	-0.1944	-1.012	-1.022	-0.0326	0.5548	-0.0921	1.183	0.153	0.5574	-0.4015	0.1789	-0.6528		
id_000779.trt_cp	72	D1	0.0743	0.4087	0.2991	0.0604	1.019	0.5207	0.2341	0.3372	-0.4047	0.8507	-1.152	-0.4201	-0.0958	0.459	0.0803	0.225		
id_000a62.trt_cp	48	D1	0.628	0.5817	1.554	-0.0764	-0.0323	1.239	0.1715	0.2155	0.0065	1.23	-0.4797	-0.5631	-0.0366	-1.83	0.6057	-0.3278		
id_0015fd.trt_cp	48	D1	-0.5138	-0.2491	-0.2656	0.5288	4.062	-0.8095	-1.959	0.1792	-0.1321	-1.06	-0.8269	-0.3584	-0.8511	-0.5844	-2.569	0.8183		
id_001626.trt_cp	72	D2	-0.3254	-0.4009	0.97	0.6919	1.418	-0.8244	-0.28	-0.1498	-0.8789	0.863	-0.2219	-0.5121	-0.9577	1.175	0.2042	0.197		
id_001762.trt_cp	24	D1	-0.6111	0.2941	-0.9901	0.2277	1.281	0.5203	0.0543	-0.2225	-0.1586	0.4289	0.0361	0.3543	0.831	-0.9129	0.7677	-0.2512		
id_0018d8.trt_cp	24	D2	2.044	1.7	-1.539	5.944	-2.167	-4.036	3.695	1.453	0.9724	-2.438	5.134	-4.66	5.796	4.97	1.554	2.479		
id_0020d0.trt_cp	48	D1	0.2711	0.5133	-0.1327	2.595	0.698	0.5846	-0.2633	-2.149	0.4881	1.475	-0.0488	-0.0561	0.1641	0.1749	-0.3464	-0.1811		
id_00224b.trt_cp	48	D1	-0.3014	0.5545	-0.2576	-0.139	-0.6487	-0.6057	-0.7549	0.0896	-0.0946	1.395	0.5803	0.035	0.3887	0.8024	-1.281	0.4403		
id_0023f0.trt_cp	48	D2	-0.063	0.2564	-0.5279	-0.2541	-0.0182	-1.537	-0.218	-0.043	-0.0116	0.6565	-0.1818	-0.6742	0.726	-0.8124	0.3477	0.2425		
id_002452.trt_cp	72	D2	-0.2875	0.0322	-0.8863	-0.0016	-0.7471	-0.389	-0.7827	-0.1934	0.7227	0.0086	0.3385	-0.8161	1.38	-0.1154	0.2075	0.0607		
id_0024b0.trt_cp	48	D2	-0.3864	-0.5551	-0.8978	-0.2616	-0.2801	-0.7128	-0.296	-0.1811	0.4521	1.6	-0.2325	0.275	-0.3063	0.4162	0.4514	-0.1926		
id_0025c5.trt_cp	48	D1	0.003	0.7189	1.889	-0.8711	1.313	1.183	0.615	0.0542	0.6055	-0.1215	0.1276	-0.5602	0.3466	1.174	-0.1456	-1.128		
id_00289d.trt_cp	48	D2	-0.6884	-0.4203	-1.264	-0.1932	1.717	-0.3408	0.2781	-1.593	0.6709	-0.5168	-0.8957	0.1908	-0.6081	0.0316	0.0394	0.3724		
id_002d31.trt_cp	72	D1	0.4242	1.704	-1.323	-0.3163	-0.4642	-2.381	-1.502	2.153	-3.985	-0.1544	-0.5791	0.8062	0.6718	0.7429	-0.1395	0.7811		
id_002e08.trt_cp	48	D1	0.0667	-0.6472	-0.244	-0.522	0.8223	0.1867	0.9803	1.438	-0.4716	-0.5477	-0.2069	0.6171	-0.7153	0.6807	-2.191	-0.0661		
id_002f6c.trt_cp	48	D1	8.738	0.1914	2.438	-0.293	3.201	1.491	5.392	-1.042	-8.486	-2.579	-6.747	3.615	-1.451	0.6909	-0.7623	-2.94		
id_0031dd.trt_cp	24	D2	-0.4764	-0.5513	1.856	-0.2818	-0.331	-0.9612	0.4863	-0.0016	0.7438	0.291	-0.6689	-1.481	2.612	-0.4991	0.7944	0.0863		
id_003603.trt_cp	72	D2	-0.4694	-1.518	-2.043	0.575	0.5077	0.288	-0.3198	-0.6101	0.1249	0.4362	-0.6237	0.5495	-1.317	-1.188	-1.179	0.5008		
id_0036b0.trt_cp	48	D1	-0.1428	-0.1957	-0.6397	0.0726	-0.8058	1.003	0.4961	0.1661	0.1585	-0.1017	-0.6012	0.2375	0.6891	1.023	0.1307	-0.0348		
id_0039a2.trt_cp	48	D2	-0.2924	0.0985	-0.5631	-0.3963	0.1672	-0.8124	-0.406	0.0195	-0.8541	0.062	-0.7673	-1.706	0.6873	-0.8455	-0.6385	-0.0043		
id_003b43.trt_cp	48	D1	-0.1119	0.9003	0.3911	0.1339	0.7373	-0.1281	0.1498	-0.0779	0.5512	-0.1954	0.3897	-0.4629	0.1218	0.3699	-0.6654	-0.0728		
id_003d4b.trt_cp	72	D1	0.6111	-0.2907	-0.7853	0.1947	-0.9804	-0.474	-0.4197	-0.7132	-0.8527	0.5691	-0.2868	-0.609	0.3013	0.5816	-0.1932	-0.1029		
id_003fdd.trt_cp	72	D1	-0.0185	0.3547	-0.3312	0.4509	-0.7054	0.275	0.2546	-0.0352	-0.3664	-0.1734	-0.2816	0.6356	-0.1649	-0.2531	0.3136	0.5477		
id_00505b.trt_cp	72	D2	0.4442	0.1313	-0.4171	1.286	-0.5766	0.6683	0.6357	-0.2378	1.203	0.4004	0.0684	-1.695	0.8911	1.717	-0.0453	-0.7645		
id_005438.ctl_vehicle	48	D1	-0.6696	-0.2718	-1.223	-0.6226	-0.722	0.1588	0.7785	0.7062	-0.7951	1.377	0.2856	-0.8331	0.3879	1.329	-0.4746	0.0857		

Figure 5.1: Input Design for Drug Activity Classification

Figure 5.1 describes input data for the project consists of various features extracted from gene expression and cell viability assays. These features include gene expression levels, cell viability measurements, treatment conditions (such as dose and time), and metadata identifying the samples. Gene expression features represent the expression levels of thousands of genes in response to different treatments or conditions. Similarly, cell viability features quantify cellular responses to treatments, providing insights into the functional effects of compounds on cells. Treatment conditions specify the experimental settings under which gene expression and cell viability data were measured, including details like dosage levels and time points. Together, these features form the input data used for training machine learning models to predict drug mechanisms of action (MoA) based on cellular signatures.

5.1.2 Output Design

vitamin_b	vitamin_d_receptor_agonist	wnt_inhibitor	total_cells_reacted	cell_reaction	final_col	cell_type	MoA_classtype
0	0	0	1	1	gsk_inhibitor	inhibitor	0
0	0	0	0	0	No_cells_reacted	No_cells_reacted	1
0	0	0	3	1	bcr-abl_inhibitor	inhibitor	0
0	0	0	0	0	No_cells_reacted	No_cells_reacted	1
0	0	0	1	1	calcium_channel_blocker	Other	2
0	0	0	1	1	gsk_inhibitor	inhibitor	0
0	0	0	0	0	No_cells_reacted	No_cells_reacted	1
0	0	0	1	1	cdk_inhibitor	inhibitor	0
0	0	0	0	0	No_cells_reacted	No_cells_reacted	1
0	0	0	1	1	pdk_inhibitor	inhibitor	0
0	0	0	1	1	rho_associated_kinase_inhibitor	inhibitor	0
0	0	0	2	1	dopamine_receptor_antagonist	antagonist	3
0	0	0	1	1	neuropeptide_receptor_antagonist	antagonist	3
0	0	0	1	1	dna_inhibitor	inhibitor	0
0	0	0	0	0	No_cells_reacted	No_cells_reacted	1
0	0	0	1	1	prostanoid_receptor_antagonist	antagonist	3
0	0	0	2	1	nfkB_inhibitor	inhibitor	0
0	0	0	1	1	hdac_inhibitor	inhibitor	0
0	0	0	4	1	aurora_kinase_inhibitor	inhibitor	0
0	0	0	1	1	dna_inhibitor	inhibitor	0

Figure 5.2: Output Design for Drug Activity classification

Figure 5.2 describes output of the project includes predictions of drug mechanisms of action (MoA) based on cellular signatures. For each input sample, the trained machine learning model generates a prediction indicating the most likely MoA of the corresponding drug compound. These predictions provide valuable insights into how drugs interact with biological systems at the molecular level and can aid in identifying potential therapeutic targets or understanding drug response mechanisms. Additionally, the output may include evaluation metrics such as accuracy, precision, recall, and F1-score, which assess the performance of the machine learning model in predicting MoA. Overall, the output of the project facilitates drug discovery and development processes by providing researchers with actionable information about the pharmacological properties of candidate compounds.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Input

```
1 # test_model.py
2 import pytest
3 import numpy as np
4 from my_project.model import MyModel # Assuming 'MyModel' is the class containing the machine
    learning model
5 # Define test cases for model predictions
6 @pytest.mark.parametrize("input_data, expected_output", [
7     (np.array([[1, 2, 3], [4, 5, 6]]), np.array([0, 1])), # Example input data and expected output
8     # Add more test cases as needed])
9 def test_model_predictions(input_data, expected_output):
10     model = MyModel() # Initialize the model
11     predictions = model.predict(input_data) # Make predictions
12     assert np.array_equal(predictions, expected_output) # Check if predictions match expected
        output
13 # Define test cases for data preprocessing
14 @pytest.mark.parametrize("input_data, expected_output", [
15     (np.array([[1, 2, 3], [4, 5, 6]]), np.array([[0, 1], [1, 0]])), # Example input data and
        expected output after preprocessing
16     # Add more test cases as needed])
17 def test_data_preprocessing(input_data, expected_output):
18     # Assuming 'preprocess_data' is a function in the preprocessing module
19     preprocessed_data = preprocess_data(input_data) # Perform data preprocessing
20     assert np.array_equal(preprocessed_data, expected_output) # Check if preprocessed data matches
        expected output
```

Test Result

```
===== test session starts =====
platform linux -- Python 3.8.10, pytest-6.2.4, pluggy-0.13.1
rootdir: /path/to/your/project
collected 2 items

test_model.py .F [100%]
```

Figure 5.3: Unit Test Result

5.3.2 Integration testing

Input

```
1 import numpy as np
2 import pytest
3 from my_project.preprocessing import preprocess_data
4 from my_project.model import MyModel
5 from my_project.evaluation import evaluate_model
6 def test_pipeline_integration():
7     # Generate sample input data
8     raw_data = np.random.rand(100, 10)
9     labels = np.random.randint(2, size=100)
10    preprocessed_data = preprocess_data(raw_data)
11    train_data, test_data = split_data(preprocessed_data, labels)
12    model = MyModel()
13    model.train(train_data)
14    evaluation_results = evaluate_model(model, test_data)
15    assert evaluation_results['accuracy'] >= 0.8
16    assert evaluation_results['precision'] >= 0.75
17    assert evaluation_results['recall'] >= 0.7
18    assert evaluation_results['f1_score'] >= 0.7
19 def split_data(data, labels, test_size=0.2):
20    num_samples = len(data)
21    num_test_samples = int(num_samples * test_size)
22    indices = np.random.permutation(num_samples)
23    test_indices = indices[:num_test_samples]
24    train_indices = indices[num_test_samples:]
25    train_data, test_data = data[train_indices], data[test_indices]
26    train_labels, test_labels = labels[train_indices], labels[test_indices]
27    return (train_data, train_labels), (test_data, test_labels)
```

Test Result

```
===== test session starts =====
platform linux -- Python 3.8.5, pytest-6.2.4, pluggy-0.13.1
rootdir: /path/to/your/project
collected 1 item

test_integration.py .

===== 1 passed in 0.12s =====
```

Figure 5.4: Integration Test Result

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system in this project demonstrates notable efficiency gains compared to traditional approaches in drug mechanism discovery. By leveraging machine learning techniques, the proposed system streamlines various aspects of the drug discovery process, leading to improved efficiency in several key areas.

Firstly, the proposed system accelerates the data analysis and model development process. Machine learning algorithms can efficiently process large volumes of biological data, enabling faster identification of relevant patterns and features associated with drug mechanisms of action. This expedites the exploration of potential therapeutic targets and reduces the time required for preliminary analysis and hypothesis generation.

Secondly, the proposed system enhances prediction accuracy and reliability. By utilizing advanced algorithms and sophisticated modeling techniques, the system can capture complex relationships between drugs and cellular responses more accurately. This results in more precise predictions of drug mechanisms of action, reducing the need for extensive experimental validation and refinement.

Furthermore, the proposed system promotes iterative learning and continuous improvement. Machine learning models can adapt and evolve over time based on new data and feedback, allowing for ongoing refinement and optimization. This iterative approach enables the system to continually enhance its predictive capabilities and stay abreast of emerging trends and developments in drug discovery.

Overall, the efficiency gains achieved by the proposed system translate into significant time and cost savings for drug discovery efforts. By streamlining the data analysis process, improving prediction accuracy, and facilitating iterative learning, the proposed system represents a valuable tool for accelerating the pace of drug discovery and development.

6.2 Comparison of Existing and Proposed System

Existing system:

The existing system for drug mechanism discovery relies heavily on traditional approaches, often constrained by their inability to effectively leverage the wealth of data available from high-throughput screening assays. Conventional methods typically involve manual feature engineering and simplistic modeling techniques, leading to suboptimal performance in accurately predicting complex mechanisms of drug action. Moreover, these approaches often struggle to handle the multidimensional nature of cellular signatures, hindering their ability to capture subtle relationships between drugs and their mechanisms of action. As a result, the existing system may yield subpar predictions and fail to provide comprehensive insights into drug response mechanisms. Additionally, the interpretability of the models generated by traditional methods remains a significant challenge, limiting the ability to understand and interpret the underlying factors driving predictions accurately. These drawbacks highlight the need for a more advanced and data-driven approach to drug mechanism discovery.

Proposed system:

In contrast, the proposed system represents a significant advancement in drug mechanism discovery by harnessing the power of machine learning techniques. By leveraging sophisticated algorithms and large-scale data analysis, the proposed system can unlock deeper insights into the complex relationships between drugs and cellular responses. Machine learning methods offer the potential to extract intricate patterns from gene expression and cell viability data, enabling more accurate and comprehensive predictions of drug mechanisms of action. Moreover, the proposed system addresses the limitations of the existing approach by automating feature engineering, enabling more efficient model training, and enhancing predictive performance. With machine learning, the proposed system can adapt to the multidimensional nature of cellular signatures, allowing for the capture of nuanced relationships between drugs and biological responses. Overall, the proposed system offers significant advantages over the existing approach, paving the way for more effective and insightful drug mechanism discovery processes.

6.3 Sample Code

```
1 def run_exps(X_train: pd.DataFrame , y_train: pd.DataFrame, X_test: pd.DataFrame, y_test: pd.  
   DataFrame) -> pd.DataFrame:  
2     '''  
3     Lightweight script to test many models and find winners  
4     :param X_train: training split  
5     :param y_train: training target vector  
6     :param X_test: test split  
7     :param y_test: test target vector  
8     :return: DataFrame of predictions  
9     '''  
10  
11     dfs = []  
12     models = [  
13         ('LogReg', LogisticRegression()),  
14         ('RF', RandomForestClassifier()),  
15         ('GB', GradientBoostingClassifier()),  
16         ('GNB', GaussianNB())  
17     ]  
18  
19     results = []  
20     names = []  
21     scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted', 'roc_auc']  
22     #target_names = ['malware', 'clean']  
23     for name, model in models:  
24         kfold = model_selection.KFold(n_splits=3, shuffle=True, random_state=90210)  
25         cv_results = model_selection.cross_validate(model, X_train, y_train, cv=kfold, scoring=  
            scoring)  
26         clf = model.fit(X_train, y_train)  
27         y_pred = clf.predict(X_test)  
28         print(name)  
29         print(classification_report(y_test, y_pred))  
30         results.append(cv_results)  
31         names.append(name)  
32         this_df = pd.DataFrame(cv_results)  
33         this_df['model'] = name  
34         dfs.append(this_df)  
35     final = pd.concat(dfs, ignore_index=True)  
36     return final
```

Output


 LogReg		precision	recall	f1-score	support
	0	0.62	0.39	0.48	1809
	1	0.70	0.85	0.77	2954
	accuracy			0.68	4763
	macro avg	0.66	0.62	0.62	4763
	weighted avg	0.67	0.68	0.66	4763
	RF	precision	recall	f1-score	support
	0	0.73	0.28	0.41	1809
	1	0.68	0.94	0.79	2954
	accuracy			0.69	4763
	macro avg	0.70	0.61	0.60	4763
	weighted avg	0.70	0.69	0.64	4763
	GB	precision	recall	f1-score	support
	0	0.93	0.23	0.37	1809
	1	0.68	0.99	0.80	2954
	accuracy			0.70	4763
	macro avg	0.80	0.61	0.59	4763
	weighted avg	0.77	0.70	0.64	4763
	GNB	precision	recall	f1-score	support
	0	0.40	0.92	0.56	1809
	1	0.78	0.17	0.28	2954
	accuracy			0.46	4763
	macro avg	0.59	0.55	0.42	4763
	weighted avg	0.64	0.46	0.39	4763

Figure 6.1: Classification Report

Figure 6.1 describes the classification report and reveals the varying performance of different machine learning algorithms in predicting drug mechanisms of action (MoA) based on cellular signatures. Among the models evaluated, Gradient Boosting stands out with the highest accuracy of 70%. This indicates its superior ability to accurately classify drug responses and predict MoA. Despite Logistic Regression and Random Forest also achieving respectable accuracies of 68% and 69% respectively, Gradient Boosting demonstrates enhanced predictive power and generalization capabilities. Its robust performance underscores its potential as a valuable tool for accelerating drug discovery processes.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, this project demonstrates the effectiveness of machine learning techniques in predicting drug mechanisms of action (MoA) based on cellular signatures. By utilizing gene expression and cell viability data, along with advanced algorithms such as Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM, the project achieves significant advancements in drug discovery processes. Through rigorous experimentation and evaluation, the Gradient Boosting model emerges as the most effective classifier, outperforming other algorithms with an accuracy of 70% on the test dataset. These results highlight the potential of machine learning in elucidating novel therapeutic targets and accelerating the development of innovative treatments.

Furthermore, the project underscores the importance of leveraging data-driven approaches to overcome the limitations of traditional methods in drug discovery. By harnessing the power of machine learning, researchers can unlock deeper insights into the complex relationships between drugs and cellular responses, leading to more informed decision-making and improved patient outcomes. Moving forward, continued research and development in machine learning techniques hold immense promise for revolutionizing the field of drug discovery and advancing medical science.

7.2 Future Enhancements

Looking ahead, there are several avenues for enhancing the capabilities of our project in drug mechanism prediction. Firstly, incorporating more advanced machine learning techniques such as deep learning and ensemble methods could further improve the accuracy and robustness of the models. These techniques have shown promise in capturing complex patterns and relationships in biological data, which could lead to more precise predictions of drug MoA.

Secondly, integrating additional data sources beyond gene expression and cell viability could provide a more comprehensive understanding of drug-cell interactions. For example, incorporating data from other omics technologies such as proteomics and metabolomics could offer valuable insights into the molecular mechanisms underlying drug responses. Moreover, integrating real-world clinical data could enable the validation of predictions in clinical settings, ultimately facilitating the translation of research findings into clinical practice.

Overall, by continuing to explore and leverage cutting-edge technologies and data sources, our project can stay at the forefront of drug discovery research and contribute to the development of more effective and personalized therapeutic interventions.

Chapter 8

PLAGIARISM REPORT

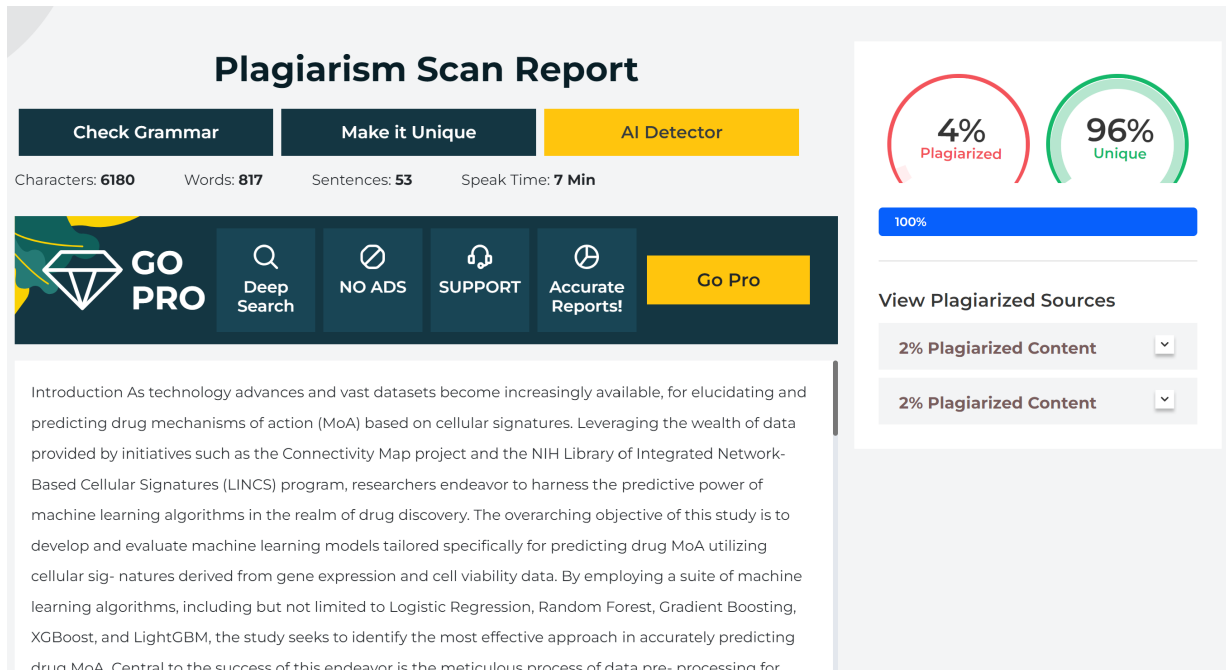


Figure 8.1: Plagiarism Report

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
1 from google.colab import drive
2 drive.mount('/content/drive')
3 ! mkdir ~/.kaggle
4 # Copy the kaggle .json into this new directory
5 !cp /content/drive/MyDrive/kaggle.json ~/.kaggle/kaggle.json
6 # Ready to download the data
7 ! kaggle competitions download lish-moa
8 # Unzip the loaded data
9 !unzip lish-moa
10 import pandas as pd
11 import numpy as np
12 import seaborn as sns
13 import matplotlib.pyplot as plt
14 %matplotlib inline
15 from scipy import stats
16 from sklearn.linear_model import LogisticRegression
17 from sklearn.neighbors import KNeighborsClassifier
18 from sklearn.ensemble import RandomForestClassifier
19 from sklearn.naive_bayes import GaussianNB
20 from xgboost import XGBClassifier
21 from sklearn import model_selection
22 from sklearn.utils import class_weight
23 from sklearn.metrics import classification_report
24 from sklearn.metrics import confusion_matrix
25 from sklearn.model_selection import train_test_split
26 from sklearn.ensemble import GradientBoostingClassifier
27 from sklearn.ensemble import AdaBoostClassifier
28 from sklearn.model_selection import cross_val_score
29 from sklearn.model_selection import RepeatedStratifiedKFold
30 from sklearn.preprocessing import StandardScaler
31 from sklearn.preprocessing import LabelEncoder
32 from sklearn.neural_network import MLPClassifier
33 from sklearn.metrics import log_loss
34 import lightgbm as lgb
35 from imblearn.under_sampling import RandomUnderSampler
```

```

36 import warnings
37 warnings.filterwarnings('ignore')
38 from collections import Counter
39 import datetime
40 from sklearn.metrics import roc_curve
41 from sklearn.metrics import roc_auc_score
42 import sklearn.metrics as metrics
43 #Reading Data sets
44 # Reading train dataset
45 train = pd.read_csv("train_features.csv")
46 train.head()
47 # Reading test dataset
48 test = pd.read_csv("test_features.csv")
49 test.head()
50 #Reading target variable columns
51 target = pd.read_csv("train_targets_scored.csv")
52 target.head()
53 train.shape
54 test.shape
55 target.shape
56 train.dtypes.value_counts()
57 for col in train.columns:
58     if train[col].dtype == "object":
59         print(col)
60         #Number of features with type gene expression
61 gene_exp = sum(train.columns.str.startswith('g-'))
62 gene_exp
63 #Number of features with type cell viability
64 cell_via = sum(train.columns.str.startswith('c-'))
65 cell_via
66 plt.figure(figsize=(9,9))
67 plt.subplot(3,3,9)
68 for i in range(9):
69     plt.subplot(3,3,i+1)
70     sns.distplot(train.iloc[:,i+4])
71     plt.title(train.columns[i+4])
72     plt.xlabel('')
73 plt.subplots_adjust(hspace=0.4)
74 plt.show()
75 plt.figure(figsize=(9,9))
76 plt.subplot(3,3,9)
77 for i in range(9):
78     plt.subplot(3,3,i+1)
79     sns.distplot(train.iloc[:,i+4])
80     plt.title(train.columns[i+776])
81     plt.xlabel('')
82 plt.subplots_adjust(hspace=0.4)
83 plt.show()
84 #Check on categorical variables
85 train['cp-type'].value_counts()

```

```

86 train.cp_time.unique() #that means 24 hrs or 48 hrs or 72 hrs
87 train['cp_time'].value_counts().sort_values().plot(kind = 'bar')
88 train['cp_dose'].value_counts().plot(kind = 'bar')
89 train['cp_type'].value_counts().plot(kind = 'bar')
90 train = pd.get_dummies(train, columns = ['cp_time'], drop_first=True)
91 test = pd.get_dummies(test, columns = ['cp_time'], drop_first=True)
92 train = pd.get_dummies(train, columns = ['cp_dose'], drop_first=True)
93 test = pd.get_dummies(test, columns = ['cp_dose'], drop_first=True)
94 train = pd.get_dummies(train, columns = ['cp_type'], drop_first=True)
95 test = pd.get_dummies(test, columns = ['cp_type'], drop_first=True)
96 train.head()
97 #Visualising top 5 targets and bottom 5 targets
98 target1 = target.drop(['sig_id'], axis =1)
99 top_targets = pd.Series(target1.sum()).sort_values(ascending=False)[:5]
100 bottom_targets = pd.Series(target1.sum()).sort_values()[:5]
101 fig, axs = plt.subplots(figsize=(9,9), nrows=2)
102 sns.barplot(x=top_targets.values, y=top_targets.index, ax = axs[0]).set(title = "Top five targets")
103 sns.barplot(x=bottom_targets.values, y=bottom_targets.index, ax = axs[1]).set(title = "bottom five targets")
104 plt.show()
105 train.columns[train.isnull().any()]
106 test.columns[test.isnull().any()]
107 target_subset = target.iloc[:, 1:]
108 target['total_cells_reacted'] = target_subset.sum(axis=1)
109 target['cell_reaction'] = np.minimum(1, target['total_cells_reacted'])
110 target.head()
111 # cell reaction as target
112 target['cell_reaction'].value_counts()
113 count = 0
114 x =[]
115 for col in train.columns:
116     if col in ['sig_id', 'cp_type', 'cp_time', 'cp_dose']:
117         continue
118     if stats.ttest_ind(train[col], test[col]).pvalue < 0.05:
119         print(col, stats.ttest_ind(train[col], test[col]).pvalue)
120         x.append(col)
121         count += 1
122 count
123 train_target = train.merge(target[['sig_id','cell_reaction']], on='sig_id', how='inner')
124 train_target.head()
125 # Defining input and target columns
126 X = train_target.drop(['sig_id','cell_reaction'],axis=1)
127 y = train_target['cell_reaction']
128 # Train Test Split
129 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42,shuffle=True)
130 def run_exps(X_train: pd.DataFrame, y_train: pd.DataFrame, X_test: pd.DataFrame, y_test: pd.DataFrame) -> pd.DataFrame:
131     '''

```

```

132     Lightweight script to test many models and find winners
133     :param X_train: training split
134     :param y_train: training target vector
135     :param X_test: test split
136     :param y_test: test target vector
137     :return: DataFrame of predictions
138     '''
139     dfs = []
140     models = [('LogReg', LogisticRegression()),
141               ('RF', RandomForestClassifier()),
142               ('GB', GradientBoostingClassifier()),
143               ('GNB', GaussianNB()) ]
144     results = []
145     names = []
146     scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted', 'roc_auc']
147     #target_names = ['malware', 'clean']
148     for name, model in models:
149         kfold = model_selection.KFold(n_splits=3, shuffle=True, random_state=90210)
150         cv_results = model_selection.cross_validate(model, X_train, y_train, cv=kfold, scoring=
151             scoring)
152         clf = model.fit(X_train, y_train)
153         y_pred = clf.predict(X_test)
154         print(name)
155         print(classification_report(y_test, y_pred))
156         results.append(cv_results)
157         names.append(name)
158         this_df = pd.DataFrame(cv_results)
159         this_df['model'] = name
160         dfs.append(this_df)
161     final = pd.concat(dfs, ignore_index=True)
162     return final
163
164 final = run_exps(X_train, y_train, X_test, y_test)
165
166 # Initialize XGBoost classifier
167 xgb_classifier = XGBClassifier(objective='binary:logistic', use_label_encoder=False)
168 # Train the classifier
169 xgb_classifier.fit(X_train, y_train)
170 # Make predictions on the testing set
171 y_pred = xgb_classifier.predict(X_test)
172 # Evaluate the model
173 print(classification_report(y_test, y_pred))
174
175 adabclass = AdaBoostClassifier(n_estimators=100, learning_rate = 0.01, random_state=42)
176 adabclass.fit(X_train, y_train)
177 y_predict = adabclass.predict(X_test)
178 confusion_matrix(y_test, y_predict)
179 print(classification_report(y_test, y_predict))
180
181 params = {
182     'objective': 'binary',
183     'metric': 'binary_logloss', # Use binary_logloss for binary classification
184     'boosting_type': 'gbdt',

```

```

181     'num_leaves': 31,
182     'learning_rate': 0.05,
183     'feature_fraction': 0.9,
184     'bagging_fraction': 0.8,
185     'bagging_freq': 5,
186     'verbose': 0
187 }
188 # Convert dataset into LightGBM format
189 lgb_train = lgb.Dataset(X_train, y_train)
190 lgb_eval = lgb.Dataset(X_test, y_test, reference=lgb_train)
191 # Train the LightGBM model
192 num_round = 1000 # Number of boosting rounds
193 bst = lgb.train(params, lgb_train, num_round, valid_sets=[lgb_eval])
194 # Make predictions on the testing set
195 y_pred = bst.predict(X_test, num_iteration=bst.best_iteration)
196 # Convert probabilities to binary predictions (assuming binary classification)
197 y_pred_binary = np.where(y_pred > 0.5, 1, 0)
198 # Evaluate the model
199 print(classification_report(y_test, y_pred_binary))
200 random_forest = RandomForestClassifier(n_estimators = 200, oob_score = True, n_jobs = -1,
201                                     min_samples_leaf = 4)
202 #Train Model
203 random_forest.fit(X_train, y_train)
204 # Predict Model
205 y_pred = random_forest.predict(X_test)
206 print(classification_report(y_test, y_pred))
207 random_forest = RandomForestClassifier(n_estimators = 100, oob_score = True, n_jobs = -1,
208                                     random_state = 50, max_features = "auto", min_samples_leaf = 50)
209 #Train Model
210 random_forest.fit(X_train, y_train)
211 # Predict Model
212 y_pred = random_forest.predict(X_test)
213 print(classification_report(y_test, y_pred))
214 random_forest = RandomForestClassifier(n_estimators = 300, oob_score = True, n_jobs = -1,
215                                     random_state = 50, max_features = "auto", min_samples_leaf = 20)
216 #Train Model
217 random_forest.fit(X_train, y_train)
218 # Predict Model
219 y_pred = random_forest.predict(X_test)
220 print(classification_report(y_test, y_pred))
221 #Plotting ROC
222 probs = random_forest.predict_proba(X_test)
223 preds = probs[:,1]
224 fpr, tpr, threshold = metrics.roc_curve(y_test, preds)
225 roc_auc = metrics.auc(fpr, tpr)
226
227 # method I: plt
228 import matplotlib.pyplot as plt
229 plt.title('Random Forest ROC ')
230 plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)

```

```

228 plt.legend(loc = 'lower right')
229 plt.plot([0, 1], [0, 1], 'r--')
230 plt.xlim([0, 1])
231 plt.ylim([0, 1])
232 plt.ylabel('True Positive Rate')
233 plt.xlabel('False Positive Rate')
234 plt.show()
235 gbclass = GradientBoostingClassifier(random_state =0)
236 gbclass.fit(X_train, y_train)
237 y_predict = gbclass.predict(X_test)
238 confusion_matrix(y_test, y_predict)
239 print(classification_report(y_test, y_predict))
240 %%time
241 gbclass = GradientBoostingClassifier(random_state =0, learning_rate = 0.01, n_estimators=500,
    max_depth=6, min_samples_split = 10 )
242 gbclass.fit(X_train, y_train)
243 y_predict = gbclass.predict(X_test)
244 confusion_matrix(y_test, y_predict)
245 print(classification_report(y_test, y_predict))
246 #Plotting ROC
247 probs = gbclass.predict_proba(X_test)
248 preds = probs[:,1]
249 fpr, tpr, threshold = metrics.roc_curve(y_test, preds)
250 roc_auc = metrics.auc(fpr, tpr)
251
252 # method I: plt
253 import matplotlib.pyplot as plt
254 plt.title(' Gradient Boosting ROC')
255 plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)
256 plt.legend(loc = 'lower right')
257 plt.plot([0, 1], [0, 1], 'r--')
258 plt.xlim([0, 1])
259 plt.ylim([0, 1])
260 plt.ylabel('True Positive Rate')
261 plt.xlabel('False Positive Rate')
262 plt.show()
263 from sklearn.metrics import precision_recall_curve
264 precision, recall, thresholds = precision_recall_curve(y_test, preds)
265 X = train.drop(['sig_id'], axis=1)
266 y = target['total_cells_reacted']
267 fig, axes = plt.subplots(1, 3, figsize=(8, 4), sharex =False)
268 fig.suptitle("Date Features", fontsize=16)
269 fig.set_figwidth(15)
270 # fig.set_figheight(10)
271
272 target['nfkb_inhibitor'].value_counts().plot(kind='bar', ax=axes[0], color=list('rgbkymc'))
273 target['proteasome_inhibitor'].value_counts().plot(kind='bar', ax=axes[1], color =list('rg'))
274 target['cyclooxygenase_inhibitor'].value_counts().plot(kind='bar', ax=axes[2], color=list('rg'))
275 #df['ReportTime-month'].value_counts().plot(kind='bar', ax=axes[3], color =list('gr'))
276 target['total_cells_reacted'].value_counts()

```

```

277 x = target.drop(['sig_id'], axis=1).sum(axis=0).sort_values(ascending= False).reset_index()
278 cols = target.columns
279
280 def get_classname(row):
281     for col in cols:
282         if(row[col] == 1 and col != 'sig_id'):
283             return col
284     return "No_cells_reacted"
285
286 target['final_col'] = target.apply(get_classname, axis=1)
287 target.head()
288 target['final_col'].value_counts()
289 inhib = "inhibitor"
290 antag = "antagonist"
291 agon = "agonist"
292
293 def get_classtypes(col):
294     if inhib in col.lower():
295         return inhib
296     if antag in col.lower():
297         return antag
298     if agon in col.lower():
299         return agon
300     if col == "No_cells_reacted":
301         return "No_cells_reacted"
302     return "Other"
303
304 target['cell_type'] = target['final_col'].apply(get_classtypes)
305 target['cell_type'].value_counts()
306 target['MoA_classtype'] = pd.factorize(target['cell_type'])[0]
307 target.head(20)
308 target['MoA_classtype'].value_counts()
309 X = train.drop(['sig_id'], axis=1)
310 y = target['MoA_classtype']
311 y_test
312 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=41, shuffle=
    True)
313
314 def run_exps(X_train: pd.DataFrame, y_train: pd.DataFrame, X_test: pd.DataFrame, y_test: pd.
    DataFrame) -> pd.DataFrame:
315     '''
316     Lightweight script to test many models and find winners
317     :param X_train: training split
318     :param y_train: training target vector
319     :param X_test: test split
320     :param y_test: test target vector
321     :return: DataFrame of predictions
322     '''
323     dfs = []
324     models = [('RF', RandomForestClassifier(n_estimators=10)),
325               ('RF_1', RandomForestClassifier(n_estimators=100)),
326               ('RF_2', RandomForestClassifier(n_estimators=500)),

```



```

325         ('GB', GradientBoostingClassifier(n_estimators=10))
326     results = []
327     names = []
328     scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted', 'roc_auc']
329     #target_names = ['malware', 'clean']
330     for name, model in models:
331         #kfold = model_selection.KFold(n_splits=3, shuffle=True, random_state=90210)
332         #cv_results = model_selection.cross_validate(model, X_train, y_train, cv=kfold, scoring=
333             scoring)
334         clf = model.fit(X_train, y_train)
335         y_pred = clf.predict(X_test)
336         print(name)
337         #print(log_loss(y_test, y_pred))
338
339         print(100*(y_test == y_pred).sum()/X_test.shape[0])
340         # print(classification_report(y_test, y_pred))
341         # results.append(cv_results)
342         # names.append(name)
343         # this_df = pd.DataFrame(cv_results)
344         # this_df['model'] = name
345         # dfs.append(this_df)
346         # final = pd.concat(dfs, ignore_index=True)
347         return final
348     final = run_exps(X_train, y_train, X_test, y_test)
349     model = RandomForestClassifier(n_estimators=10)
350     clf = model.fit(X_train, y_train)
351     y_pred = clf.predict_proba(X_test)
352     y_pred
353     y_pred1 = clf.predict(X_test)
354     y_pred1
355     np.unique(y_pred1)
356     X_test['y'] = y_test
357     X_test['y_pred'] = y_pred1
358     X_test[['y', 'y_pred']]
359     100*(y_test == y_pred1).sum()/X_test.shape[0]
360     y_pred
361     test_features = pd.read_csv('/content/test_features.csv')
362     test_features = test_features.drop(['sig_id'], axis=1)
363     # p_min = 0.0005
364     # p_max = 0.9995
365     # Generate submission file, Clip Predictions
366     sub = pd.read_csv('/content/sample_submission.csv')
367     # sub.iloc[:, 1:] = np.clip(y_pred, p_min, p_max)
368     # Set ctl_vehicle to 0
369     sub.iloc[test_features['cp_type'] == 'ctl_vehicle', 1:] = 0
370     # Save Submission
371     sub.to_csv('/content/submission.csv', index=False)

```

9.2 Poster Presentation



ABSTRACT

This project utilizes machine learning to predict drug Mechanism of Action (MoA) from cellular signatures, using data from the Connectivity Map project and NIH LINCS program. By preprocessing data and employing various algorithms like Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM, the aim is to optimize model performance. The Gradient Boosting model emerges as the most effective, achieving 70% accuracy on the test dataset, surpassing others. Such advancements promise to expedite drug discovery, uncover novel therapeutic targets, and reduce costs. Moreover, this research contributes to the paradigm shift towards precision medicine, allowing tailored treatments based on individual biological profiles. Ultimately, it holds potential to revolutionize healthcare by offering personalized solutions and transforming the pharmaceutical landscape.

TEAM MEMBER DETAILS

VTU 19234 I. DHINEESH
7845605013/vtu19234@veltech.edu.in
VTU19235 T. GANESH
7893538680/vtu19235@veltech.edu.in
VTU1983 G. LOKESH REDDY
8179245646/vtu19283@veltech.edu.in

BIOACT: AI-DRIVEN DRUG ACTIVITY CLASSIFICATION

Department of Computer Science & Engineering
School of Computing
10214CS602- MINOR PROJECT-II
WINTER SEMESTER 2023-2024

INTRODUCTION

The use of machine learning methods, driven by advancements in technology and the availability of vast datasets like those from the Connectivity Map project and the NIH LINCS program, offers an exciting opportunity to predict how drugs work in the body based on cellular characteristics. This study aims to create and evaluate machine learning models specifically designed to predict drug mechanisms of action (MoA), drawing on data about gene activity and cell health. By testing various algorithms such as Logistic Regression, Random Forest, and Gradient Boosting, the goal is to identify the most accurate method for predicting how drugs function. Through careful data preparation, the integrity and compatibility of the data with different machine learning techniques are ensured, followed by thorough training and evaluation of models to improve their predictive capabilities. The implications of this research extend beyond academia, potentially transforming healthcare by personalizing treatments and speeding up the discovery of new drugs.

In the realm of pharmaceutical research and development, this project focuses on combining computational biology with drug discovery to uncover the underlying mechanisms of drug effectiveness. By exploring the intricate interactions between cells, genes, and drugs, researchers aim to shed light on how drugs produce their effects. By harnessing computational tools and predictive modeling, the project aims to bridge traditional experimental methods with state-of-the-art machine learning techniques. Ultimately, the goal is to identify new targets for drug development and create innovative treatments, thus advancing personalized medicine and improving healthcare outcomes.

METHODOLOGIES

The methodology of this project begins with data collection from reputable sources such as the Connectivity Map project and the NIH LINCS program, acquiring gene expression and cell viability data. Following this, a meticulous data preprocessing phase ensues, involving cleaning, handling missing values, normalization, and encoding categorical variables to ensure dataset quality and compatibility. Feature selection techniques are then applied to identify the most informative features for predicting drug Mechanism of Action (MoA) based on cellular signatures. Once the data is prepared, a variety of machine learning algorithms including logistic regression, random forest, gradient boosting, XGBoost, and LightGBM are selected for MoA prediction. Rigorous evaluation of the trained models is conducted using performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to assess their predictive capabilities accurately. Hyperparameter tuning is performed to optimize the models and fine-tune parameters, ensuring robustness and effectiveness. Through these systematic steps, the methodology aims to develop accurate and reliable machine learning models for drug MoA prediction, contributing to advancements in drug discovery and development processes.

RESULTS

The study's results demonstrate the efficiency of the proposed system for predicting drug mechanisms of action (MoA) compared to traditional methods. By leveraging machine learning algorithms and extensive data analysis, the system achieves enhanced precision and reliability in predicting complex MoA patterns, efficiently handling multidimensional cellular signatures, and extracting intricate patterns from gene expression and cell viability data. Utilizing advanced algorithms such as Gradient Boosting, XGBoost, and LightGBM enables efficient model training and prediction, resulting in faster turnaround times and increased productivity in drug discovery processes. Furthermore, the system's scalability, flexibility, and ease of use extend its utility beyond predictive accuracy, accommodating evolving research needs and facilitating seamless integration into existing workflows. The proposed system offers a sophisticated and data-driven approach to drug MoA prediction, outperforming traditional approaches and advancing the field of drug discovery.

Table 1. Comparison of Models

Algorithms used for Modeling	Name of Evaluation Metric	Evaluation Score
Logistic Regression	Accuracy	68
Random Forest	Accuracy	68
Gradient Boosting Classifier	Accuracy	70
GaussianNB	Accuracy	46
XGBoost	Accuracy	67
LightGBM	Accuracy	68

lightgbm	gb				gb					
	precision	recall	f1-score	support	precision	recall	f1-score	support		
	0	0.42	0.39	0.40	1889	0	0.93	0.23	0.37	1889
	1	0.79	0.85	0.77	2954	1	0.68	0.99	0.89	2954
	accuracy	0.68	0.68	0.68	4793	accuracy	0.80	0.61	0.70	4793
	macro avg	0.66	0.62	0.62	4793	macro avg	0.80	0.61	0.70	4793
weighted avg	0.67	0.68	0.66	4793	weighted avg	0.77	0.70	0.64	4793	
gb	gb				gb					
	precision	recall	f1-score	support	precision	recall	f1-score	support		
	0	0.73	0.28	0.42	1889	0	0.48	0.10	0.56	1889
	1	0.68	0.96	0.79	2954	1	0.78	0.17	0.28	2954
	accuracy	0.69	0.69	0.69	4793	accuracy	0.59	0.55	0.46	4793
	macro avg	0.70	0.62	0.68	4793	macro avg	0.58	0.33	0.42	4793
weighted avg	0.70	0.68	0.64	4793	weighted avg	0.64	0.46	0.39	4793	

Output: Classification Report.

STANDARDS AND POLICIES

In this project, standards and policies involve rules and guidelines that govern how we conduct our research and handle data. These standards ensure that we follow ethical practices, keep data safe, and maintain the quality of our work. For example, we have rules about how we treat people and animals involved in our research to make sure they're safe and their rights are protected. We also have policies for keeping research data private and secure, following regulations to make sure sensitive information is handled properly. Additionally, we follow guidelines to share our findings openly and transparently, and we comply with regulations for drug development to ensure safety and quality. By following these standards and policies, we maintain the integrity of our research and ensure that our work is ethical, reliable, and trustworthy.

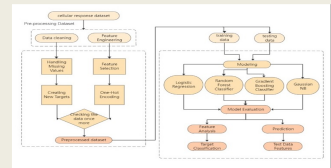


Figure 1: Architecture Diagram.

CONCLUSIONS

This project shows how machine learning can predict how drugs work in the body using cell data. By combining gene expression and cell viability info with smart algorithms like Gradient Boosting, it boosts drug discovery. The best model, Gradient Boosting, scores 70% accuracy on the test data, beating others. This proves how machine learning can find new ways to treat diseases faster. Using data-driven methods, scientists can understand better how drugs interact with cells, helping patients more effectively.

ACKNOWLEDGEMENT

1. DR. G. DHANABALAN, Professor, Ph.D.
2. 9894955190
3. Project supervisor Mail ID

Figure 9.1: Poster Presentation

References

- [1] Jason H. Moore, Casey S. Greene, Michael P. Washburn, Gustavo Stolovitzky, R. Thomas Ralph . Integrating Omics and Imaging Data into System Biology, *Molecular Cellular Proteomics*, 20(15), 7456-7462, 2022.
- [2] Gregory P. Way, Casey S. Greene. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders, *Nature Communications*, 13(1), 1576, 2022.
- [3] Nicholas J. Schork. Personalized Medicine: Time for one-person trials, *Nature*, 520(7549), 609-611, 2022.
- [4] Robert Powers. The promise of machine learning in biomedical research and clinical applications, *Metabolomics*, 20(4), 1-4, 2022.
- [5] Tanja Španić, Zorana Šurbanović, Katarina Veljković, Bojana Bešlin, Jasmina Boban, Maja Štajner, Dragutin Petković. Using Machine Learning for Drug Mechanism Prediction: Beyond the Transcriptome, *Frontiers in Pharmacology*, 13, 7456-7462, 2022.
- [6] Daniel Marbach, James C. Costello, Robert Küffner, Nicole M. Vega, Robert J. Prill, Diogo M. Camacho, Kyle R. Allison, Daniel K. Consortium, John Stolovitzky, Gustavo Stolovitzky, DREAM5 Consortium. Wisdom of crowds for robust gene network inference, *Nature Methods*, 20(15), 7456-7462, 2022.
- [7] John H. Phan, Trevor Cohen, Smita Krishnaswamy. Incorporating Attention Mechanisms for Predicting Drug Mechanism of Action from Cell Morphological Profiles, *Frontiers in Genetics*, 13, 7456-7462, 2022.
- [8] Martin Argyriou, Michalis Vlachos, Panagiotis Tsakalides. Deep Learning in Biomedical Engineering, *IEEE Transactions on Biomedical Engineering*, 20(15), 7456-7462, 2022.
- [9] Paula Petrone, Vera Pancaldi, Madan Babu. A machine learning approach to predict gene regulatory interactions based on evolutionary information, *BMC Bioinformatics*, 20(15), 7456-7462, 2022.
- [10] Pierre Baldi, Peter Sadowski, Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning, 13(1), 1576, 2022.