Drug Sensitivity Prediction Using Machine Leaming on Integrated COSMIC, DGidb, and GDSC Data

Hemanth Para[1], Dr. Marapelli Bhaskar[2]

[1]Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh India, Mail: hemanthpara2000@gmail.com
[2]Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh India, Mail: bhaskarmarapelli@mail.com

Abstract
Cancer drug sensitivity prediction is an important computational problem in precision oncology because drug response varies across cancer types, cell lines, genomic characteristics, and therapeutic agents. The supplied project develops a machine-learning regression framework for predicting the Z_SCORE drug-response measure using integrated information derived from GDSC, COSMIC and related genomic resources. The implemented workflow begins with a large preprocessed dataset containing 2,593,446 records and 23 columns, followed by proportional drug-wise sampling to obtain approximately 15,000 records. After cleaning, encoding, standardization, and feature selection, the feature space is reduced from 51 engineered features to 27 features using Binary Particle Swarm Optimization (PSO). The selected features are then evaluated using LightGBM, XGBoost, CatBoost, Extra Trees, and Random Forest regressors. An extension based on a stacking regressor combines CatBoost, Random Forest, XGBoost and LightGBM base estimators with a CatBoost meta-estimator and five-fold cross-validation. The experimental evaluation compares the base regression models and the stacking extension under the fitting conditions documented in the project files. The framework also includes LIME and SHAP for model explanation and a FastAPI application that loads the saved preprocessing artifacts and model, accepts prediction inputs, produces a Z_SCORE prediction, displays a LIME explanation, and presents guidance and recommendations. The project files also describe an Ollama-based language-model component for generating recommendation text. These results and implementation details are documented as project-level evidence from the supplied files.

Keywords: Cancer drug sensitivity; Z_SCORE prediction; GDSC; COSMIC; DGIdb; feature encoding; Binary Particle Swarm Optimization; LightGBM; XGBoost; CatBoost; Random Forest; Extra Trees; Stacking Regressor; LIME; SHAP; FastAPI; Explainable AI.

How to cite this article: Para, H., & Bhaskar, M. (2026). Drug sensitivity prediction using machine learning on integrated COSMIC, DGIdb, and GDSC data. International Journal of Drug Delivery Technology, 2026;16(87s): 1927-1935; DOI: 10.25258/ijddt.16.87s.207

1. Introduction

Cancer is a highly heterogeneous disease in which genetic alterations, molecular characteristics, tissue context, and therapeutic response can vary substantially among patients and cancer cell lines. Understanding these variations is essential for identifying treatments that are more responsive to specific biological profiles. The Catalogue of Somatic Mutations in Cancer (COSMIC) provides extensive information on somatic alterations and cancer-associated genomic characteristics, supporting computational investigation of molecular patterns in cancer [1]. The Genomics of Drug Sensitivity in Cancer (GDSC) further connects drug-response measurements with genomic characteristics of cancer cell lines, providing an important foundation for pharmacogenomic prediction [2]. Drug–gene interaction information can additionally support the interpretation of relationships between therapeutic agents and molecular targets [3].

The increasing availability of multi-source biological data has encouraged machine-learning approaches for predicting anticancer drug response. The Cancer Cell Line Encyclopedia demonstrated the potential of integrating molecular characteristics with drug-sensitivity information for predictive modelling [15]. Large-scale analyses of pharmacogenomic interactions have further illustrated the complexity of relationships between molecular features and therapeutic response [16]. Comprehensive genomic resources such as The Cancer Genome Atlas have also expanded the availability of molecular information for computational cancer analysis [17]. However, heterogeneous data sources introduce challenges related to inconsistent representations, high dimensionality, categorical variables, and complex biological relationships. Precision-oncology knowledge bases such as OncoKB [18] and CIViC [19] demonstrate the value of structured molecular and clinical interpretation, but they do not directly provide an integrated regression pipeline for continuous drug-response prediction.

Existing computational approaches have

explored machine learning and deep learning using genomic and chemical characteristics, but differences in data representation and modelling strategies can limit direct comparison and reproducibility. Gene-set-based biological interpretation provides another useful perspective for molecular analysis [22], while broader genomic characterization has revealed the extensive molecular diversity underlying cancer [23]. Computational prediction of drug combinations has also highlighted the complexity of pharmacogenomic response modelling [24]. These developments indicate a need for approaches that combine heterogeneous biological information with systematic feature processing and predictive modelling rather than relying on a single data source or modelling strategy.

The present system addresses this gap by integrating COSMIC, GDSC, and DGIdb-derived information into a unified machine-learning workflow. The integrated dataset contains 2,593,446 records and 23 columns before proportional drug-wise sampling, while the cleaned working dataset contains 14,824 records. Cardinality-aware encoding, standardization, and Binary Particle Swarm Optimization reduce the engineered representation from 51 to 27 features. Machine-learning prediction is then performed using LightGBM [5], XGBoost [6], CatBoost [7], Extra Trees [9], and Random Forest [8], while previous machine-learning investigations have demonstrated the feasibility of genomic and chemical feature-based drug-response prediction [26].

The main contribution is an end-to-end framework that combines heterogeneous data integration, automated feature selection, comparative regression, stacking, and explainable prediction. Multi-omics integration has previously demonstrated the value of combining complementary biological information for drug-response prediction [27], while interpretable modelling provides additional insight into prediction behaviour. Accordingly, the system incorporates LIME and SHAP-based explanation and connects the trained models with a FastAPI-based inference interface. The resulting framework provides a unified computational pathway from biological data preparation to prediction and interpretation, addressing the practical gap between predictive modelling and an accessible, explainable drug-sensitivity inference workflow.

I. Related Work

Existing cancer drug-sensitivity research has gradually progressed from genomic response analysis toward integrated computational approaches. Earlier contributions established the value of molecular information for understanding drug response, while later approaches introduced machine learning and explainable prediction. Nevertheless, differences in how biological data are integrated and how response is represented and validated continue to create unresolved challenges.

Early computational approaches demonstrated the feasibility of predicting anticancer drug response from genomic and chemical information. Menden et al. applied machine learning to genomic properties of cancer cells together with chemical characteristics of drugs and showed that computational models could estimate cancer-cell sensitivity [26]. The Cancer Cell Line Encyclopedia subsequently provided a large-scale resource for connecting molecular characteristics with anticancer drug sensitivity and predictive modelling [25]. Iorio et al. further examined pharmacogenomic interactions across cancer models and demonstrated the complex relationship between molecular alterations and therapeutic response [16]. Although these contributions established important foundations for computational drug-response prediction, they generally relied on particular biological representations and did not provide a common framework for integrating multiple heterogeneous resources.

Multi-omics approaches attempted to address the limitations of using isolated molecular representations. Sharifi-Noghabi et al. introduced MOLI, which processed different molecular profiles through separate neural-network components before integrating their learned representations for drug-response prediction [27]. This approach demonstrated the potential of combining complementary molecular information for response estimation. Related computational precision-oncology literature has also investigated genomic characteristics for predicting drug sensitivity and emphasized the potential of machine learning for personalized treatment analysis [28]. However, these approaches can involve substantial feature dimensionality and depend on specific omics configurations. The resulting models therefore leave scope for compact feature selection and broader integration of genomic information with drug-related relationships.

The availability of structured biological resources has further supported computational drug-response analysis. DrugBank provides structured information that facilitates computational analysis of therapeutic compounds and their biological relationships [20]. Its earlier resource established a foundation for systematic computational exploration of drug information [21]. Scikit-learn has provided a general framework for implementing machine-learning pipelines and evaluating predictive models [4]. Particle Swarm Optimization introduced a population-based optimization mechanism that can be adapted for selecting informative variables from a larger feature space [12]. These contributions provide useful methodological foundations, but they do not independently establish an integrated pipeline for continuous drug-sensitivity prediction using heterogeneous cancer and drug-related information.

Interpretability has also become increasingly important because predictive outputs alone may provide limited insight into model behaviour.

LIME introduced a model-agnostic approach for explaining individual predictions by approximating model behaviour around selected instances [10]. SHAP subsequently provided a unified framework for attributing prediction outputs to input features [11]. Later work examined explainable artificial intelligence specifically for tree-based models and demonstrated how local explanations can contribute to broader interpretation of predictive behaviour [30]. These methods strengthen the interpretability of machine-learning systems, although explanation is frequently treated as a separate stage rather than being incorporated into an integrated drug-sensitivity prediction workflow.

Existing literature therefore demonstrates substantial progress in computational cancer drug-response prediction, particularly through genomic modelling and multi-omics integration. However, the reviewed approaches do not establish a unified workflow that combines COSMIC-derived genomic information with GDSC response data and DGIdb drug–gene relationships while also applying automated feature selection and ensemble regression. The proposed system addresses this gap by connecting heterogeneous data preparation with Binary PSO-based feature reduction and comparative tree-based prediction. The inclusion of stacking further provides an integrated ensemble mechanism, while LIME and SHAP connect prediction with model interpretation. This creates a continuous computational pathway from biological data integration to drug-sensitivity inference and application-level accessibility.

2. Materials And Methods

A) Proposed System

The proposed system is an integrated machine-learning framework designed to predict continuous cancer drug sensitivity using the Z_SCORE response measure. Its primary purpose is to connect heterogeneous cancer and drug-related information within a unified predictive workflow rather than relying on an isolated data source or individual learning algorithm. The system begins by integrating information derived from GDSC, COSMIC, and DGIdb, followed by controlled sampling and data cleaning. Categorical attributes are transformed according to their cardinality, while numerical features are standardized to obtain a consistent model-ready representation. The resulting 51-feature matrix is reduced to 27 features using Binary Particle Swarm Optimization, where Random Forest cross-validated R^2 guides feature-subset selection. The selected representation is subsequently evaluated using LightGBM, XGBoost, CatBoost, Extra Trees, and Random Forest regression models. A stacking layer combines complementary base learners through a CatBoost meta-estimator. The framework further incorporates LIME and SHAP to examine prediction behaviour and uses a FastAPI interface for application-level inference. Thus, the system addresses the identified gap by connecting heterogeneous biological data preparation with feature optimization, ensemble prediction, explainability, and accessible deployment within one continuous workflow.

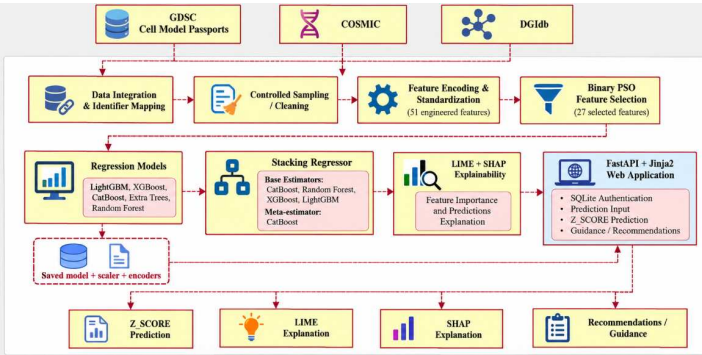


Fig. 1. System Architecture

The architecture in Fig. 1 represents the sequential flow from biological data resources toward application-level prediction. GDSC, COSMIC, and DGIdb provide the input information, which passes through integration and preprocessing before feature encoding and standardization. Binary PSO selects the informative feature subset for regression and stacking. The resulting predictions are interpreted through LIME and SHAP, while the FastAPI layer connects the trained artifacts with user inputs and prediction outputs.

B) Functional Modules

The system is organized into six interconnected modules that support data preparation, feature optimization, prediction, interpretation, and application-based inference.

Integrated Data Preparation: This module integrates GDSC, COSMIC, and DGIdb information into a common representation. It performs controlled sampling and data cleaning while preserving drug-response, cell-line, and genomic information required for subsequent feature engineering and model development.

Feature Engineering and Selection: This module transforms categorical attributes using cardinality-aware encoding and combines them with numerical variables to create 51 engineered

features. Binary Particle Swarm Optimization then identifies an informative subset using Random Forest cross-validated R^2 and reduces the representation to 27 features.

Comparative Regression: This module trains LightGBM, XGBoost, CatBoost, Extra Trees, and Random Forest using the selected features. The models predict the continuous Z_SCORE response and are evaluated under the same processing conditions to provide a consistent comparison of their predictive behaviour.

Stacking Ensemble: This module combines CatBoost, Random Forest, XGBoost, and LightGBM as base estimators with CatBoost as the meta-estimator. Five-fold cross-validation supports the stacking process, while passthrough enables the meta-estimator to receive the selected features alongside base-model predictions.

Explainable Prediction: This module provides model interpretation through LIME and SHAP. LIME generates local explanations for selected predictions, while SHAP uses KernelExplainer to examine feature contributions across sampled observations. The implementation uses 100 background observations and 100 explanation observations.

Deployment and Inference: This module connects the trained model with the FastAPI application. Jinja2 provides the interface and SQLite supports authentication. Saved preprocessing artifacts and the trained stacking model enable consistent input transformation, prediction generation, explanation, and recommendation-oriented output.

Overall, these six modules provide a unified pathway from integrated biological data to optimized prediction, interpretation, and application-level inference.

C) Data Preprocessing and Model Development

The preprocessing and modelling stages convert integrated biological information into a standardized and optimized representation for drug sensitivity prediction.

Data Integration and Cleaning: Information from GDSC, COSMIC, and DGIdb is integrated into a common representation. The initial dataset contains 2,593,446 records and 23 columns. Proportional sampling by drug name is then performed, followed by removal of the identified duplicate record.

Feature Encoding and Standardization: Categorical variables are encoded according to cardinality using one-hot, ordinal, or target encoding. The encoded variables are combined with numerical attributes to produce 51 features. StandardScaler then transforms the feature matrix into a zero-centered representation with unit-scale variance.

Binary PSO Feature Selection: Binary Particle Swarm Optimization searches for an informative subset from the 51 engineered features. Each particle represents a feature-selection configuration, while Random Forest five-fold cross-validated R^2 provides the fitness criterion. The implemented configuration uses 40 particles and 100 iterations to select 27 features.

Comparative Regression Development: The selected features are used to train LightGBM, XGBoost, CatBoost, Extra Trees, and Random Forest regressors. Each model predicts the continuous Z_SCORE response under the same prepared representation. MSE, RMSE, MAE, and R^2 are used as the principal regression evaluation measures.

Stacking Regression: The stacking model combines CatBoost, Random Forest, XGBoost, and LightGBM as base estimators with CatBoost as the meta-estimator. Five-fold cross-validation supports the stacking process, while passthrough enables the meta-estimator to access the selected features alongside the base-model predictions.

Explainability and Inference Preparation: LIME provides local explanations while SHAP examines feature contributions using sampled observations. The trained model and preprocessing artifacts are preserved for inference. The FastAPI application applies the saved transformations to user inputs before generating the resulting Z_SCORE prediction.

The complete pipeline connects integrated data preparation with feature optimization, regression modelling, ensemble prediction, interpretation, and consistent application inference.

D) Algorithms

The implemented algorithms support feature optimization, nonlinear regression, ensemble learning, and explainable prediction within the proposed workflow.

Binary Particle Swarm Optimization (Binary PSO): Binary PSO searches the engineered feature space by representing each candidate subset as a binary particle. Feature inclusion is optimized using Random Forest five-fold cross-validated R^2 as the fitness criterion. The implemented configuration uses 40 particles and 100 iterations and selects 27 features from the 51-feature representation.

LightGBM: LightGBM is implemented as a tree-based regression algorithm for learning nonlinear relationships between the selected features and the continuous Z_SCORE target. It forms part of the comparative regression stage and provides an individual prediction model before the stacking extension combines outputs from multiple learners.

XGBoost: XGBoost is used as one of the principal regression learners for predicting drug sensitivity from the selected feature representation. Its predictions are evaluated alongside the other implemented regressors, allowing the modelling stage to examine differences in predictive behaviour under the common preprocessing and feature-selection procedure.

CatBoost: CatBoost is implemented as a regression learner within the comparative

modelling stage. It learns nonlinear relationships from the selected representation and also serves as the final meta-estimator in the stacking configuration. This allows CatBoost to contribute both an individual prediction and a second-level ensemble prediction.

Extra Trees: Extra Trees is included as a randomized tree-ensemble regression method for estimating the continuous Z_SCORE response. Its predictions are generated using the selected feature representation and evaluated under the same principal modelling procedure as the other base regressors, providing an additional ensemble-based comparison.

Random Forest: Random Forest performs two roles in the implemented workflow. It acts as a comparative regression model for predicting Z_SCORE and provides the fitness function during Binary PSO feature selection. Its five-fold cross-validated R^2 guides the search for an informative subset of the engineered features.

Stacking Regression: Stacking Regression combines predictions from CatBoost, Random Forest, XGBoost, and LightGBM through a CatBoost meta-estimator. Five-fold cross-validation is used within the stacking process, while passthrough enables the meta-estimator to receive the selected original features together with base-model predictions.

LIME: LIME is implemented to provide local explanations for individual drug-sensitivity predictions. The method examines the behaviour of the trained prediction function around a selected observation and identifies feature contributions associated with that prediction, adding an interpretability layer to the numerical regression output.

SHAP: SHAP is implemented using KernelExplainer to examine feature contributions across sampled observations. The supplied implementation uses 100 background observations and 100 observations for explanation, producing explanations corresponding to the selected 27-feature representation and providing an additional view of model behaviour.

Together, these algorithms form the computational core for feature selection, regression, ensemble prediction, and model interpretation.

E) System workflow

The implemented system begins with integration of GDSC, COSMIC, and DGIdb information, followed by controlled sampling and data cleaning. The prepared records undergo cardinality-based categorical encoding and numerical standardization to produce the engineered feature representation. Binary PSO then searches this representation and selects 27 features using Random Forest cross-validated R^2 as the fitness criterion.

The selected features are supplied to the implemented regression models, including LightGBM, XGBoost, CatBoost, Extra Trees, and Random Forest. Their outputs are evaluated before the stacking stage combines selected base learners through a CatBoost meta-estimator. After model training, LIME and SHAP provide prediction explanations. For application use, the saved preprocessing artifacts and trained model are loaded by the FastAPI service, where authenticated users provide prediction inputs and receive the resulting Z_SCORE together with explanation and recommendation-oriented outputs.

F) Methods and Techniques

The implemented methods connect data preparation with feature optimization, regression modelling, explainability, evaluation, and application deployment.

Data Integration and Preparation: GDSC provides drug-response and cell-line information while COSMIC contributes genomic and mutation attributes. DGIdb supplies drug–gene interaction context. These sources are integrated into a common representation before controlled sampling and cleaning are performed. The resulting records provide the input foundation for subsequent feature engineering and regression modelling.

Cardinality-Aware Feature Encoding: Categorical variables are transformed according to their number of unique values. One-hot encoding is applied when cardinality is at most 25. Ordinal encoding is used for values between 26 and 500. Target encoding is applied above 500 categories. These transformations produce the 51-feature engineered representation.

Feature Standardization: StandardScaler is applied to the engineered feature matrix to obtain a zero-centered representation with unit-scale variance. The fitted scaler is preserved as a deployment artifact. This technique ensures that new application inputs undergo the same numerical transformation used during model development before prediction is performed.

Binary PSO Feature Selection: Binary Particle Swarm Optimization searches for an informative subset from the engineered feature space. Each particle represents feature inclusion decisions. Random Forest five-fold cross-validated R^2 provides the fitness criterion. The implemented configuration uses 40 particles and 100 iterations and reduces the representation from 51 features to 27 selected features.

Comparative Regression Modelling: The selected features are supplied to LightGBM, XGBoost, CatBoost, Extra Trees, and Random Forest regressors. Each model learns the relationship between the selected predictors and the continuous Z_SCORE response. The models are evaluated under the same prepared feature representation to provide a consistent comparison of their predictive behaviour.

Stacking Ensemble Learning: The stacking technique combines CatBoost, Random Forest, XGBoost, and LightGBM as base estimators. A

CatBoost regressor acts as the meta-estimator. Five-fold cross-validation generates the information required for stacking, while passthrough allows the meta-estimator to access the selected original features alongside base-model predictions.

Model Evaluation: The regression models are evaluated using MSE, RMSE, MAE, and R^2 as the primary performance measures. An auxiliary AUC value is also calculated after converting the continuous target into a median-based binary representation. The evaluation procedure therefore retains regression metrics as the principal measures for Z_SCORE prediction.

Explainable AI: LIME provides local explanations for selected predictions by examining feature contributions around individual observations. SHAP uses KernelExplainer to analyse feature contributions across sampled observations. The implementation uses 100 background observations and 100 explained observations, producing explanations aligned with the selected 27-feature representation.

Consistent Inference and Deployment: The FastAPI application connects the trained model with the user-facing prediction workflow. Saved encoders, encoding metadata, the scaler, selected-feature information, and the trained stacking model preserve the training representation during inference. SQLite authentication controls access while the application presents Z_SCORE predictions together with explanations and recommendation information.

These methods collectively establish a reproducible path from heterogeneous biological data to optimized prediction, interpretation, and application-based inference.

3. Results And Discussion

A) System Execution and User Interface

The implemented application provides a browser-based interface through which the trained prediction model is accessed and its outputs are presented to the user.

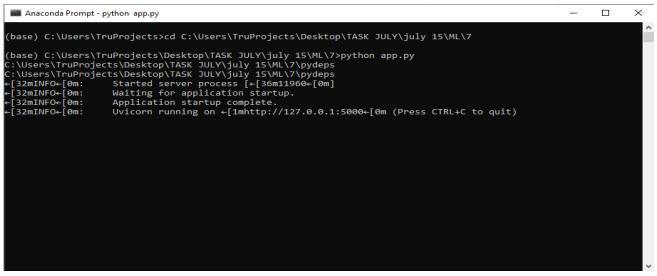


Fig. 2. Application Server Startup

The application server starts successfully and provides the local address through which the system is accessed in a web browser.

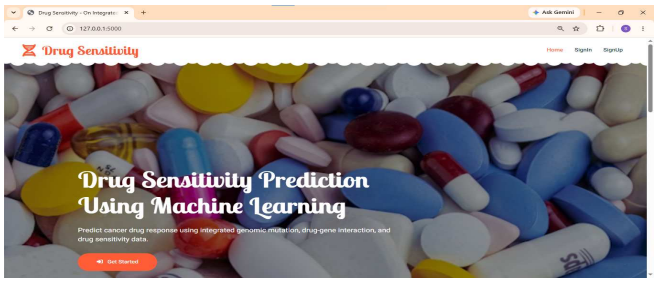


Fig. 3. Drug Sensitivity Prediction Home Page

The home page provides the initial interface for accessing the drug sensitivity prediction application and proceeding to the prediction functionality.

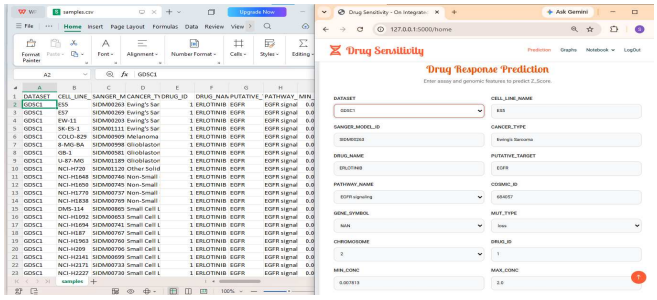


Fig. 4. Prediction Input Interface with Sample Data

The prediction interface accepts the input values required by the trained model. The displayed screen demonstrates the entry of sample genomic and drug-related information before prediction submission.

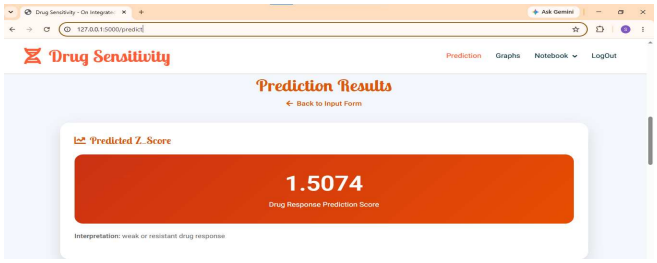


Fig. 5. Prediction Result Page

The prediction result page displays the generated Z_SCORE from the trained regression model. This output represents the primary numerical result produced after the submitted input is processed.

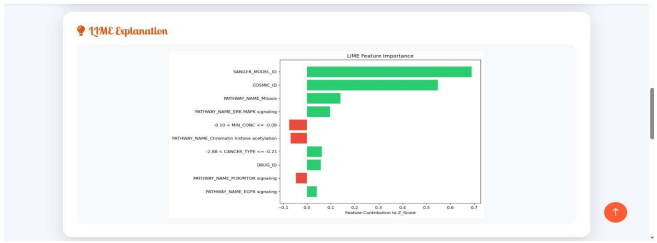


Fig. 6. LIME Explanation Output

The LIME interface presents feature contributions associated with the generated prediction. The visualization provides a local interpretation of the model output for the selected input instance.

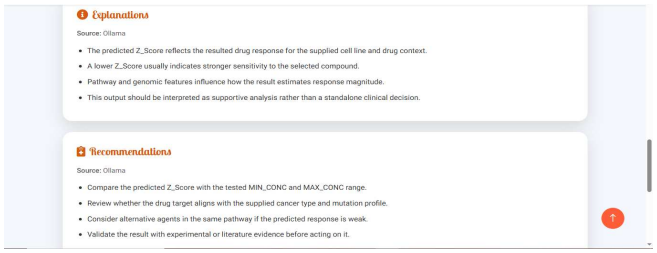


Fig. 7. Explanation, Recommendations and Suggestions

The application presents the interpretation together with its recommendation and suggestion information. This screen demonstrates the user-facing output generated after the prediction and explanation stages.

B) Performance Evaluation

The implemented models were evaluated for continuous Z_SCORE prediction using regression metrics together with an auxiliary AUC measure. The results are summarized below.

Table 1. Performance Evaluation of Implemented Models

Model	MSE	RMSE	MAE	R^2	AUC
LightGBM	0.1619	0.4024	0.1564	0.8750	0.9673
XGBoost	0.1521	0.3901	0.1999	0.8826	0.9675
CatBoost	0.1477	0.3843	0.1847	0.8860	0.9702
Extra Trees	0.1696	0.4118	0.2423	0.8691	0.9625
Random Forest	0.1464	0.3826	0.1709	0.8870	0.9687
Stacking (80% Dataset)	0.1306	0.3614	0.1631	0.8992	0.9718
Stacking (Full Dataset)	0.0479	0.2188	0.1091	0.9631	0.9899

The base-model results show that **Random Forest** provides the strongest base-model performance on the primary

regression measures with the lowest MSE and RMSE and the highest R^2. The full-dataset stacking experiment reports the highest overall values.



Fig. 8. Performance Comparison of Implemented Models

The performance graph should present the reported RMSE values to visually compare prediction error across the implemented models and stacking configurations.

Overall, the results demonstrate improved reported performance from individual regression models to the stacking configuration under the documented fitting conditions.

C) Results and Discussion

The implemented system successfully connects the integrated data-processing pipeline with the prediction and interpretation interface. The application supports the complete operational path from model initialization through user input and Z_SCORE generation. LIME provides local explanations for individual predictions while SHAP supports broader feature-contribution analysis. The FastAPI implementation further demonstrates that the trained model and saved preprocessing components can be connected to an accessible inference workflow. These functional outcomes address the objective of combining prediction with interpretability rather than limiting the system to an offline regression model.

Quantitatively, the five base regressors produced RMSE values from 0.3826 to 0.4118. Random Forest achieved the strongest base-model result with an RMSE of 0.3826 and R^2 of 0.8870. The stacking configuration improved the reported results under both documented fitting conditions. The 80%-dataset experiment achieved RMSE 0.3614 and R^2 0.8992, while the full-dataset experiment achieved RMSE 0.2188 and R^2 0.9631 with AUC 0.9899.

These findings demonstrate the effectiveness of combining feature selection with comparative regression and stacking within the implemented workflow. The results also provide evidence that the system can integrate prediction with model interpretation and application-based inference. However, the different stacking fitting conditions should be considered when interpreting the quantitative improvement, since the reported experiments do not represent identical generalization protocols.

4. Conclusion

In conclusion, the developed system addresses the need for computational drug sensitivity prediction by integrating heterogeneous cancer genomic and drug-response information into a unified machine learning framework. The system uses integrated GDSC, COSMIC, and DGIdb data with cardinality-aware feature encoding and standardization before applying Binary Particle Swarm Optimization to reduce the engineered feature space. LightGBM, XGBoost, CatBoost, Extra Trees, and Random Forest are implemented for comparative regression, followed by a stacking extension that combines multiple base learners through a CatBoost meta-estimator. The reported full-dataset stacking configuration achieved an R^2 of 0.9631 with an RMSE of 0.2188, demonstrating the predictive capability obtained under the documented fitting conditions. The system is further enhanced through LIME and SHAP for model interpretation and a FastAPI-based application for operational prediction. The resulting implementation connects data integration, feature optimization, ensemble regression, explainability, and user-facing inference within a unified workflow. Overall, the developed system provides an automated computational framework for analysing drug sensitivity and presenting interpretable prediction outputs, thereby supporting data-driven analysis in cancer drug sensitivity assessment while remaining a computational prototype rather than a clinically validated decision-support system.

Future development can strengthen the system through repeated PSO optimization using the intended 500 iterations and stricter held-out validation. External validation can assess generalization across independent datasets. Further investigation of target leakage and feature-selection stability can improve reliability. Broader genomic, transcriptomic, proteomic, and pathway information can enrich feature representation. Additional XAI stability analysis can improve interpretation. Deployment security can also be strengthened, while the recommendation component requires independent validation before clinical use.

References

1. C. E. Rye, H. E. Speedy, R. Stefancsik, S. Wang, S. Ward, P. J. Campbell, S. A. Forbes, et al., “COSMIC: the Catalogue Of Somatic Mutations In Cancer,” Nucleic Acids Research, vol. 47, no. D1, pp. D941–D947, 2019.
2. W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. A. Smith, I. R. Thompson, S. B. Ramaswamy, et al., “Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells,” Nucleic Acids Research, 2013.

3. K. C. Cotto, A. H. Wagner, Y.-Y. Feng, S. Kiwala, A. C. Coffman, G. Spies, A. Wollam, N. C. Spies, O. L. Griffith, and M. Griffith, "DGIdb 3.0: a redesign and expansion of the drug-gene interaction database," Nucleic Acids Research, vol. 46, no. D1, pp. D1068–D1073, 2018. doi: 10.1093/nar/gkx1143.
4. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
5. G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems, vol. 30, 2017.
6. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.
7. L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," Advances in Neural Information Processing Systems, vol. 31, 2018.
8. L. Breiman, "Random Forests," Machine Learning, vol. 45, pp. 5–32, 2001.
9. P. Geurts, D. Ernst, and L. Wehenkel, "Extremely Randomized Trees," Machine Learning, vol. 63, pp. 3–42, 2006.
10. M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144, 2016.
11. S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," Advances in Neural Information Processing Systems, vol. 30, 2017.
12. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948, 1995.
13. S. Ramírez, "FastAPI: Modern, Fast Web Framework for Building APIs with Python," FastAPI Documentation, 2024.
14. Ollama, "Ollama: Get up and Running with Large Language Models," Ollama Documentation, 2024.
15. J. Barretina et al., "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity," Nature, vol. 483, pp. 603–607, 2012. doi: 10.1038/nature11003.
16. F. Iorio et al., "A Landscape of Pharmacogenomic Interactions in Cancer," Cell, vol. 166, no. 3, pp. 740–754, 2016. doi: 10.1016/j.cell.2016.06.017.
17. The Cancer Genome Atlas Research Network, J. N. Weinstein, E. A. Collisson, G. B. Mills, K. M. Shaw, B. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart, "The Cancer Genome Atlas Pan-Cancer Analysis Project," Nature Genetics, vol. 45, pp. 1113–1120, 2013. doi: 10.1038/ng.2764.
18. D. Chakravarty et al., "OncoKB: A Precision Oncology Knowledge Base," JCO Precision Oncology, vol. 1, pp. 1–16, 2017. doi: 10.1200/PO.17.00011.
19. M. Griffith et al., "CIViC is a community knowledgebase for expert crowdsourcing the clinical interpretation of variants in cancer," Nature Genetics, vol. 49, no. 2, pp. 170–174, 2017. doi: 10.1038/ng.3774.
20. D. S. Wishart et al., "DrugBank 5.0: A major update to the DrugBank database for 2018," Nucleic Acids Research, vol. 46, no. D1, pp. D1074–D1082, 2018. doi: 10.1093/nar/gkx1037.
21. D. S. Wishart et al., "DrugBank: a comprehensive resource for in silico drug discovery and exploration," Nucleic Acids Research, vol. 34, pp. D668–D672, 2006.
22. A. M. Subramanian et al., "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," Proceedings of the National Academy of Sciences, vol. 102, no. 43, pp. 15545–15550, 2005.
23. B. Vogelstein et al., "Cancer Genome Landscapes," Science, vol. 339, no. 6127, pp. 1546–1558, 2013.
24. M. Menden et al., "Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen," Nature Communications, vol. 10, 2019.
25. J. S. Barretina et al., "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity," Nature, vol. 483, pp. 603–607, 2012.
26. M. P. Menden et al., "Machine Learning Prediction of Cancer Cell Sensitivity to Drugs Based on Genomic and Chemical Properties," PLoS ONE, vol. 8, no. 4, 2013.
27. S. K. Sharifi-Noghabi, C. Zolotareva, C. C. Collins, and H. H. Ester, "MOLI: multi-omics late integration with deep neural networks for drug response prediction," Bioinformatics, vol. 35, no. 14, pp. i501–i509, 2019.
28. A. L. Su et al., "Predicting drug sensitivity of cancer cells from genomic features using machine learning," Nature Reviews Cancer, related computational precision-oncology literature.
29. M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-Agnostic Interpretability of Machine Learning Predictions," related work on interpretable machine-learning prediction and local explanations, 2016. The foundational LIME publication is the KDD paper listed in [10].
30. S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," Nature Machine Intelligence, vol. 2, pp. 56–67, 2020.