Project Title: Drug Sensitivity Prediction Using Multi-Modal Data (GDSC Dataset)

Course: CS581A/481A Multi-modal ML in Biomedicine

Team Members:

- Balaji Gollapalli Venkataswamy Reddy
- Bharath Kumar
- Karthik Maganahalli Prakash

Project Description:

This project focuses on predicting the drug response (measured as LN_IC50) for cancer cell lines using multi-modal biological data from the GDSC dataset. The model uses genomic and transcriptomic features to predict how sensitive different cell lines are to various drugs.

Machine Learning Workflow:

1. Preprocessing:

- Drug-wise missing value imputation using statistical, KNN, and Random Forest methods.
- Encoded features using binary encoding, one-hot encoding, target encoding, and label encoding.

2. Model:

- XGBoost Regressor trained on 80% of the data, tested on the remaining 20%.
- Hyperparameter tuning using RandomizedSearchCV.

3. Evaluation Metrics:

- R^2 Score, RMSE, MAE

4. Interpretability:

- SHAP (SHapley Additive Explanations) to understand feature influence on predictions.

File Structure:

- `gdsc_xgboost_model.pkl` → Trained XGBoost model (saved with joblib)
- `Gdsc Project Presentation.pptx` → Presentation slides
- `STATEMENT.txt` → Academic honesty statement
- `README.txt` → Instructions to run the code
- `balaji_contribution.txt`, `bharath_contribution.txt`, `karthik_contribution.txt` → Individual contribution summaries
- `notebooks/` → Jupyter notebooks containing preprocessing, modeling, and visualization steps
- `data/` → Raw and cleaned dataset files used in the project

How to Run the Project:

1. Open the Jupyter notebook `notebooks/gdsc_prediction_pipeline.ipynb`.
2. Run all cells in order:
    - First cells load and clean the data.
    - Later cells train and evaluate the model.
    - The final cells generate visualizations and SHAP explanations.
3. Required Libraries:
    - pandas, numpy, matplotlib, seaborn, plotly
    - scikit-learn, xgboost, shap
4. To reuse the trained model:

```python
import joblib
model = joblib.load("gdsc_xgboost_model.pkl")
predictions = model.predict(X_test)
```