NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

DEPARTMENT OF BIOTECHNOLOGY

Course: BT299 Mini Project – I (EPICS based)

Title : A Machine Learning Framework for Predicting Drug-Drug Interactions Using Molecular Fingerprints in Gastrointestinal and Metabolic Drugs

Presented by :

Santhosh J K

Roll No: 123121

Faculty Mentor : Dr. Sudarshana Deepa V

1. WHY THIS PROJECT ?

Increasing Prevalence of Polypharmacy

Challenges with Traditional DDI Detection

Leveraging Machine Learning for Improved Prediction

2. OBJECTIVE OF THE PROJECT

PRIMARY GOAL

Develop a machine learning framework for predicting drug-drug interactions (DDIs) in gastrointesticinal and metabic drugs using molecular fingerprints

KEY AIMS

- Utilize molecular descriptors & fingerprints to identify DDIs
- Explore advanced machine learning models for optimal DDI prediction

EXPECTED OUTCOME

- Enhance model performance through feature engineering (e,g., drug pair feature combinations)

3. LITERATURE REVIEW

RESEARCH PAPER	FINDINGS
Davis, Dr E. "**Applications of Machine Learning Algorithms in Predicting Drug-Drug Interactions**." International Journal of Transcontinental Discoveries, ISSN (2018): 14-19. [2]	This study shows that machine learning (ML) offers a promising alternative to traditional methods for predicting drug-drug interactions (DDIs), addressing the scalability and accuracy limitations of previous approaches. By leveraging diverse data sources—such as **chemical structures, pharmacokinetics, and genomics**—ML algorithms like **support vector machines and neural network**s can significantly enhance prediction efficiency.
Wang, Jihong, Xiaodan Wang, and Yuyao Pang. "**StructNet-DDI: Molecular Structure Characterization-Based ResNet for Prediction of Drug–Drug Interactions.**" Molecules 29, no. 20 (2024): 4829. [3]	The study shows that using **SMILES structures, Morgan fingerprints**, and molecular descriptors improves DDI prediction by providing key pharmacokinetic insights. The StructNet-DDI model effectively captures molecular features, with future enhancements planned for better interpretability and drug safety applications.
Cheng, Feixiong, and Zhongming Zhao. "**Machine learning-based prediction of drug–drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties**." Journal of the American Medical Informatics Association 21, no. e2 (2014): e278-e286. [4]	This study presented the **HNAI framework** for predicting drug-drug interactions (DDIs) using drug similarities. Machine learning models, including SVM, were applied to a DrugBank dataset, **achieving an AUC of 0.67.** The framework effectively identified novel DDIs, especially with antipsychotic drugs, demonstrating a simple yet efficient approach for predicting unknown DDIs.

4. TECH STACK & TOOLS

Programming Language & Development Environment

Python – For efficient data processing, ML model development, and visualization

Jupyter Notebook - For development and experimentation

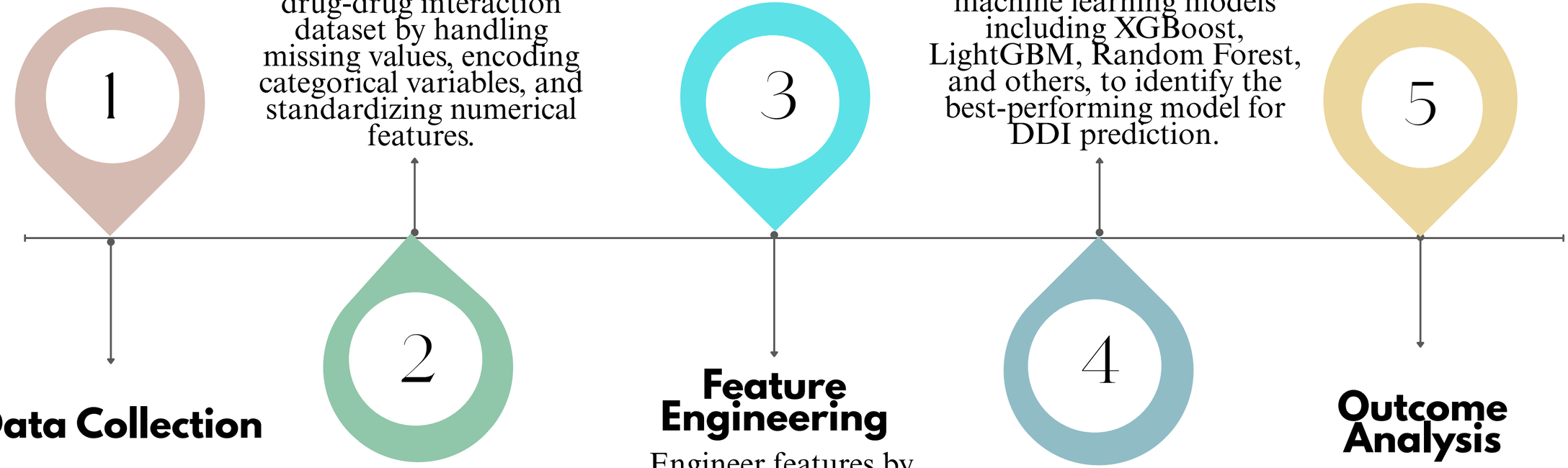
Libraries & Frameworks

- Pandas – Data manipulation and preprocessing.
- NumPy – Numerical operations and array handling.
- RDKit – Molecular structure parsing, descriptor & fingerprint generation.
- Scikit-learn – ML models, preprocessing, and evaluation metrics.
- XGBoost – Gradient boosting for high performance.
- LightGBM – Fast, efficient boosting algorithm.
- CatBoost – Boosting with categorical feature support.
- Matplotlib & Seaborne – Visualization of results.

Data Source

- DDInter – Drug interaction data
- PubChem – Drugs Molecular data
- DrugBank - Drug related informatics.

5. METHODOLOGY OVERVIEW



Data Collection

Obtained datasets containing drug interactions and extracted individual drug proprerties..

Data Preprocessing

Clean and preprocess the drug-drug interaction dataset by handling missing values, encoding categorical variables, and standardizing numerical features.

Feature Engineering

Engineer features by concatenating drug pair fingerprints, calculating absolute differences and averages of molecular descriptors to enhance model performance.

Model Training and Evaluation

Train and evaluate multiple machine learning models including XGBoost, LightGBM, Random Forest, and others, to identify the best-performing model for DDI prediction.

Outcome Analysis

Assess model performance using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, selecting the best model based on ROC-AUC.

6. DATASET OVERVIEW

Total Records: 12,220 drug-drug interaction pairs
Total Columns: 23
Target Classes (Level): 0(Low or No),1(Medium),2(High)

Feature Categories

- **Molecular Descriptors (per drug):**
  - MolecularWeight_A, MolecularWeight_B
  - XLogP_A, XLogP_B
  - TPSA_A, TPSA_B
  - Charge_A, Charge_B
  - HBondDonorCount_A, HBondDonorCount_B
  - HBondAcceptorCount_A, HBondAcceptorCount_B
  - RotatableBondCount_A, RotatableBondCount_B
  - Volume3D_A, Volume3D_B

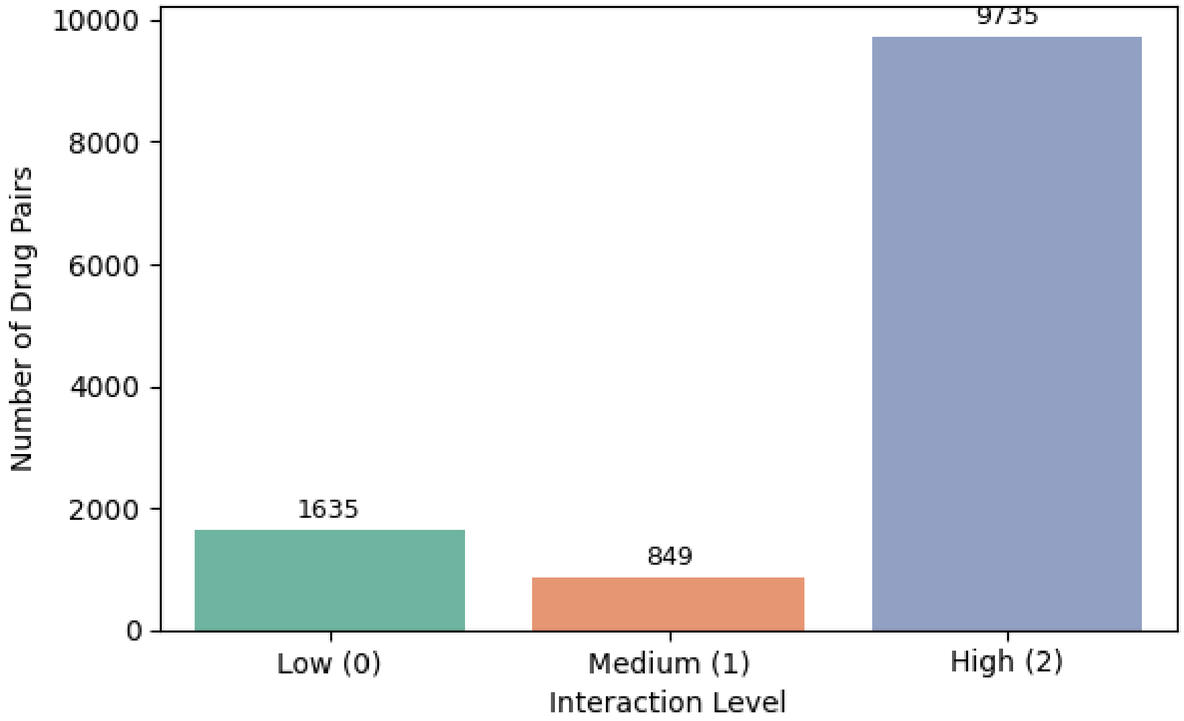
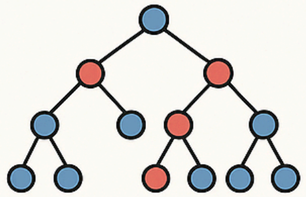


Fig 1.Class Distribution: Interaction Level

- **Structural Representation:**
  - CanonicalSMILES_A, CanonicalSMILES_B →Converted to Morgan Fingerprints (ECFP4, 1024-bit)

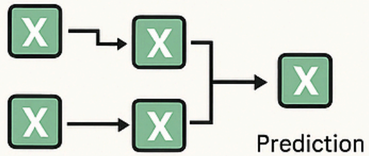
7. MODELS OVERVIEW

Random Forest



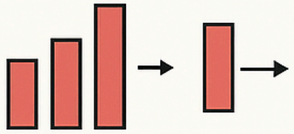
Ensemble of multiple decision trees

XGBoost



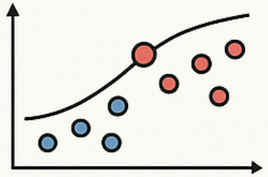
Boosted trees for regression and classificaction

LightGBM



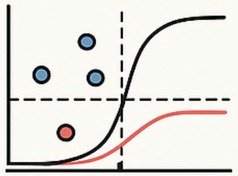
Gradient-based one-sided sampling

CatBoost



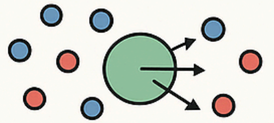
Boosted trees with categorical features

Logistic Regression



Sigmoid function for binary outcomes

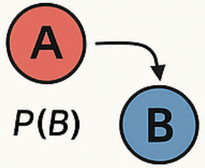
K-Nearest Neighbors



Classification

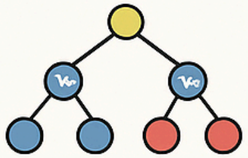
Classifies based on neighboring points

Naive Bayes



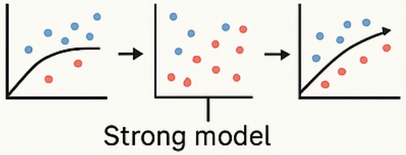
Probabilistic classifier using Bayes' theorem

Decision Tree



Tree structure for decision making

Gradient Boosting



Strong model

Combines weak models to improve

8. DATA ANALYSIS AND VISUALIZATION OF MOLECULAR FEATURES

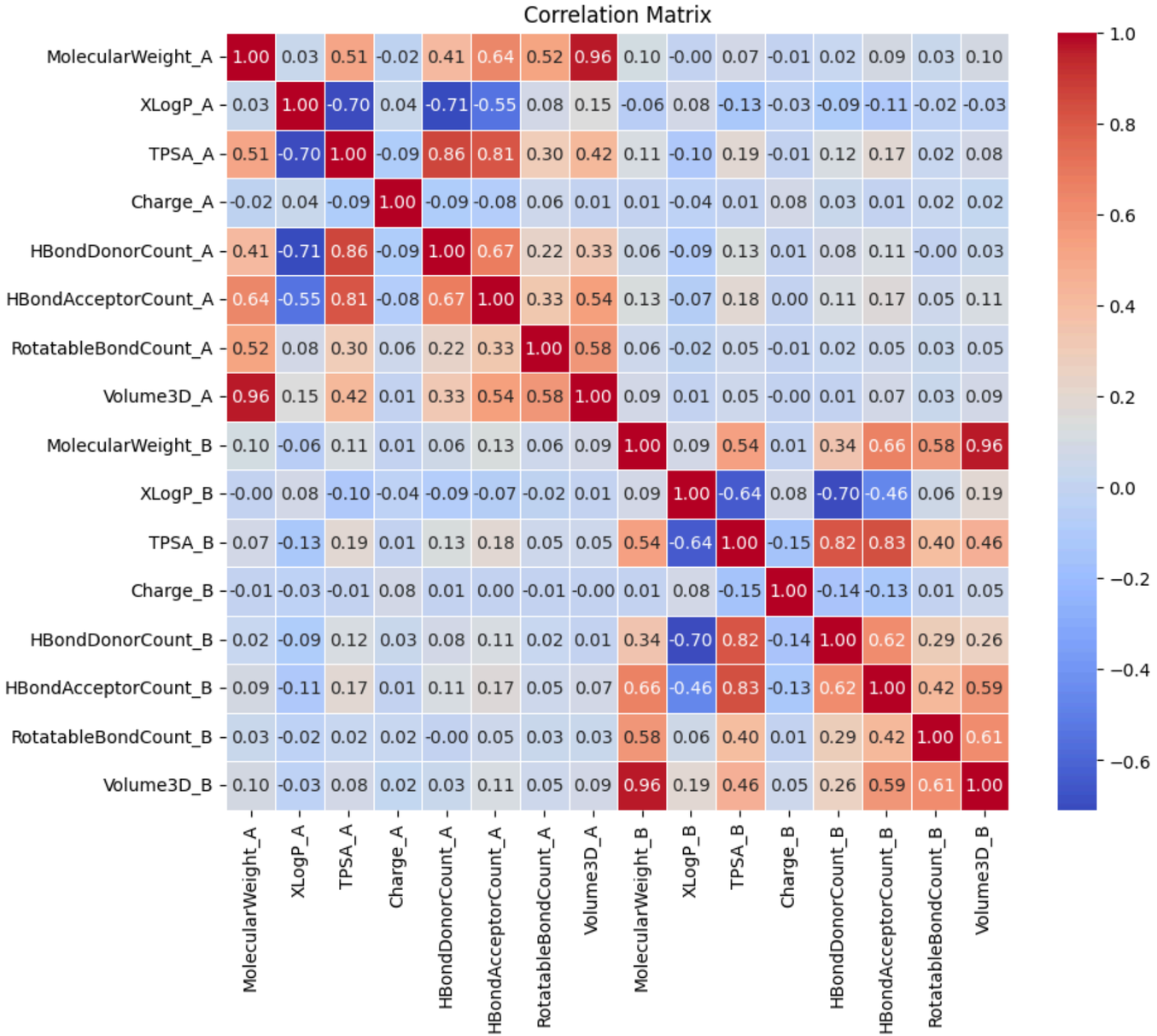


Fig 2. Heatmap: Shows the correlation between different molecular features (e.g., MolecularWeight, XLogP, TPSA) and their relationships in drug-drug interaction prediction.

8. DATA ANALYSIS AND VISUALIZATION OF MOLECULAR FEATURES

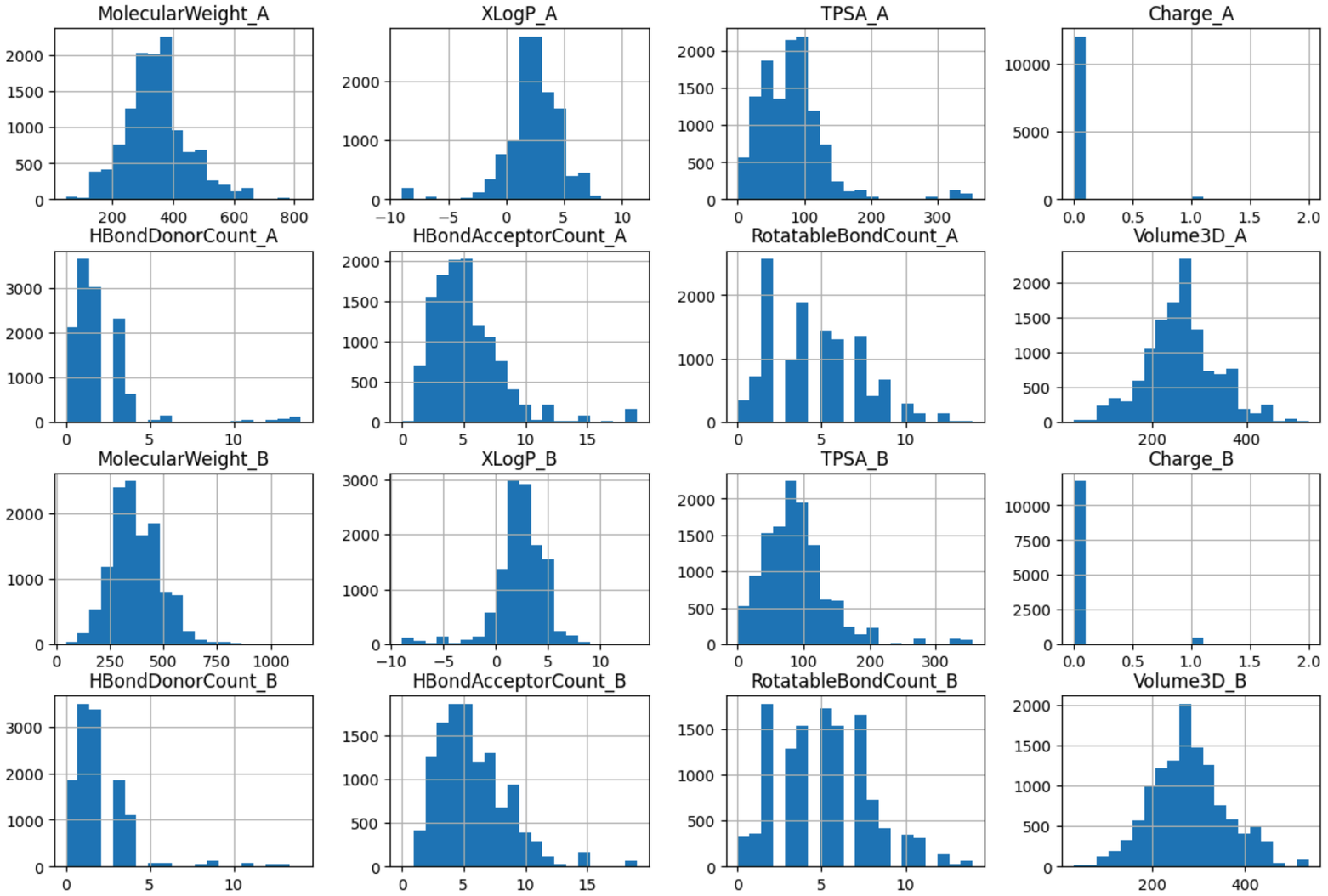


Fig3. Feature Distribution: Displays the distribution of key molecular features, providing insights into the range and variability of these features in the dataset.

8. DATA ANALYSIS AND VISUALIZATION OF MOLECULAR FEATURES

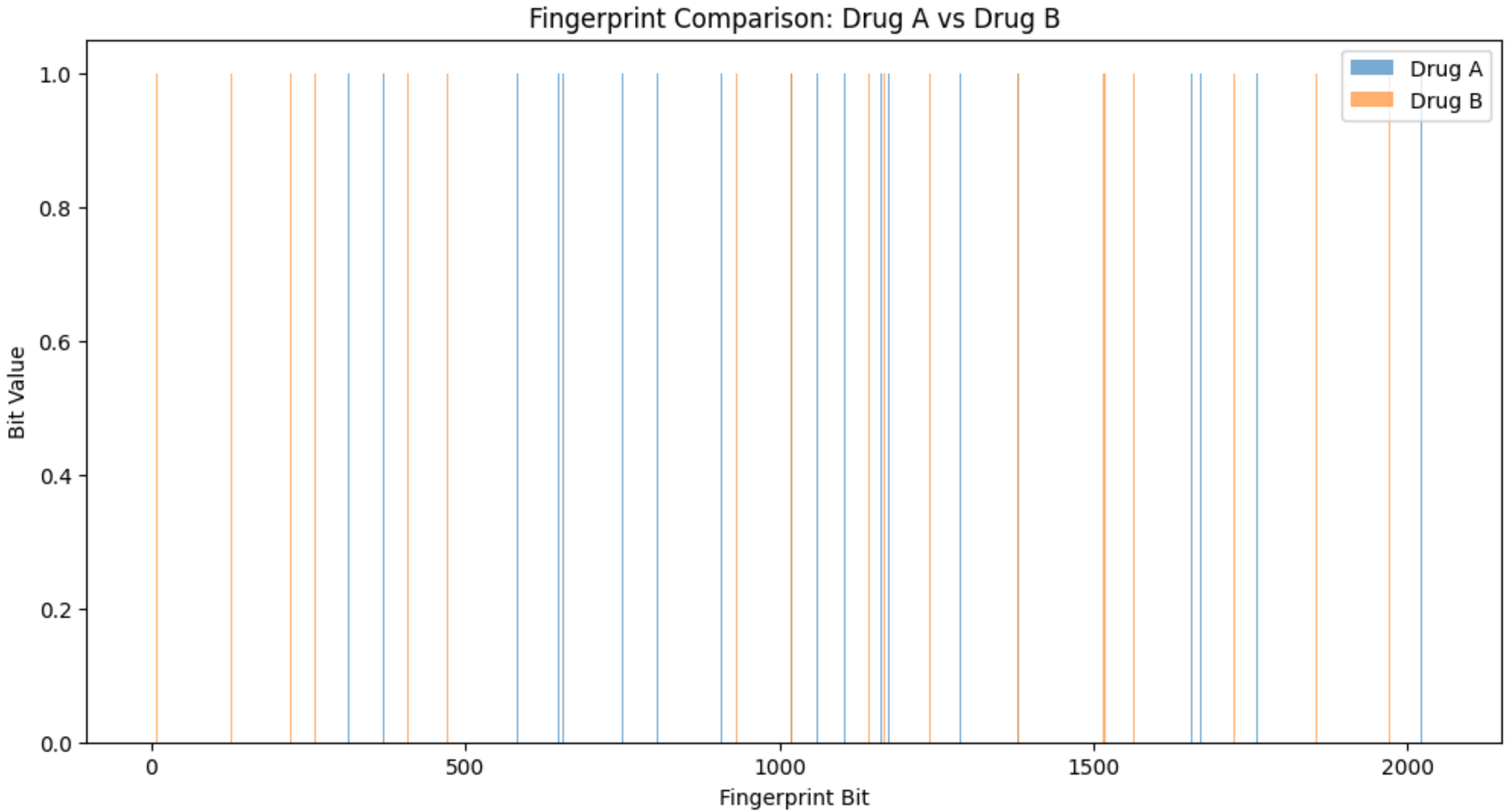


Fig 4. Fingerprint Visualization: Illustrates the binary representation of the molecular fingerprints used to characterize the drug molecules. This highlights the unique features captured in the fingerprint for prediction.

9. PIPELINE IMPLEMENTATION OVERVIEW

```python
import pandas as pd
import numpy as np
import rdkit
from rdkit import Chem, DataStructs
from rdkit.Chem import AllChem, rdFingerprintGenerator
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
import warnings

# Function to convert SMILES to Morgan Fingerprints (Handles RDKit version differences)
def smiles_to_fp(smiles, radius=2, nBits=2048):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        if hasattr(rdFingerprintGenerator, "GetMorganGenerator"):
            # New RDKit method
            generator = rdFingerprintGenerator.GetMorganGenerator(radius=radius, fpSize=nBits)
            fp = generator.GetFingerprint(mol)
        else:
            # Old RDKit method
            fp = AllChem.GetMorganFingerprintAsBitVect(mol, radius, nBits)

        arr = np.zeros((nBits,))
        DataStructs.ConvertToNumpyArray(fp, arr)
        return arr
    else:
        return np.zeros((nBits,))

# Load dataset (Ensure 'data.csv' is in the same directory)
df = pd.read_csv('data.csv')

# Convert SMILES to Fingerprints
df['FP_A'] = df['CanonicalSMILES_A'].apply(smiles_to_fp)
df['FP_B'] = df['CanonicalSMILES_B'].apply(smiles_to_fp)

# Combine Features (Concatenation, Difference, and Product of Fingerprints)
def combine_features(fp1, fp2):
    return np.concatenate((fp1, fp2, np.abs(fp1 - fp2), fp1 * fp2))

df['Features'] = df.apply(lambda row: combine_features(row['FP_A'], row['FP_B']), axis=1)

```

Fig5(a). End-to-End Model Development

This slide displays the complete code used for data preprocessing, feature extraction, model training, and evaluation. It outlines the entire pipeline, from SMILES to molecular fingerprints to the final model predictions, demonstrating how the machine learning models were implemented and evaluated for Drug-Drug Interaction prediction.

9. PIPELINE IMPLEMENTATION OVERVIEW

Fig5(b) . End-to-End Model Development

This slide displays the complete code used for data preprocessing, feature extraction, model training, and evaluation. It outlines the entire pipeline, from SMILES to molecular fingerprints to the final model predictions, demonstrating how the machine learning models were implemented and evaluated for Drug-Drug Interaction prediction.

```python
# Prepare data
X = np.vstack(df['Features'])
y = df['Level'].astype('category').cat.codes  # Convert categorical labels to numerical

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Define models
models = {
    'RandomForest': RandomForestClassifier(n_estimators=200),
    'XGBoost': XGBClassifier(use_label_encoder=False, eval_metric='logloss'),
    'LightGBM': LGBMClassifier(),
    'CatBoost': CatBoostClassifier(verbose=0),
    'LogisticRegression': LogisticRegression(max_iter=1000),
    'KNN': KNeighborsClassifier(n_neighbors=5),
    'NaiveBayes': GaussianNB(),
    'DecisionTree': DecisionTreeClassifier(max_depth=10),
    'GradientBoosting': GradientBoostingClassifier(n_estimators=200)
}

# Train & Evaluate models
results = {}
for name, model in models.items():
    print(f"Training {name}...")
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_prob = model.predict_proba(X_test) if hasattr(model, 'predict_proba') else None

    results[name] = {
        'Accuracy': accuracy_score(y_test, y_pred),
        'Precision': precision_score(y_test, y_pred, average='weighted'),
        'Recall': recall_score(y_test, y_pred, average='weighted'),
        'F1-score': f1_score(y_test, y_pred, average='weighted'),
        'ROC-AUC': roc_auc_score(y_test, y_prob, multi_class='ovr') if y_prob is not None and y_prob.shape[1] > 1 else 'N/A'
    }
    print(results)

# Print results
results_df = pd.DataFrame(results).T
print("\n 🔹 **Model Performance Metrics** 🔹")
print(results_df)
```

10. BENCHMARKING MODEL PERFORMANCE FOR DDI PREDICTION

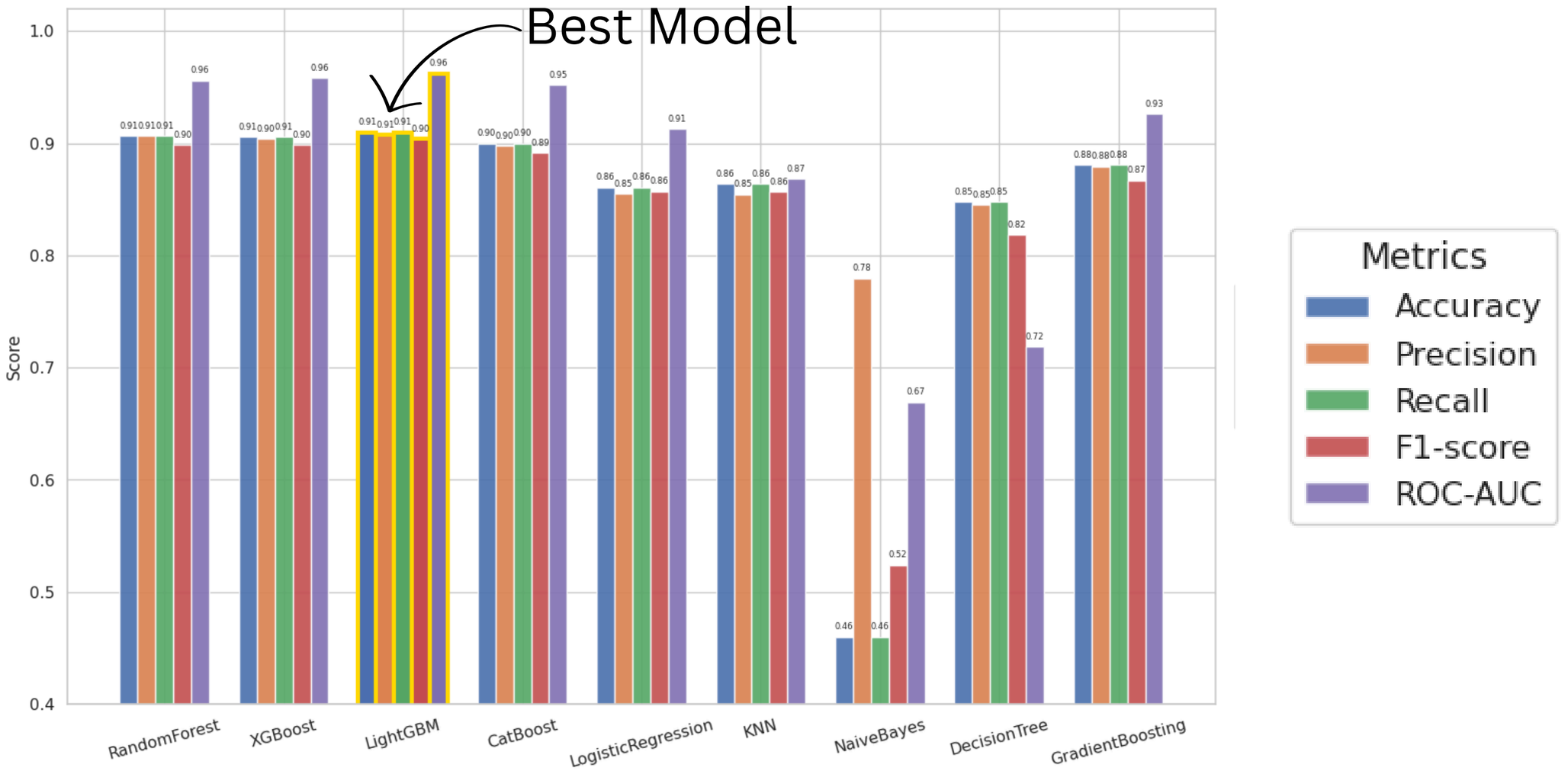


Fig6. Bar Plot with Model Results : This bar chart visualizes the performance of different models in predicting Drug-Drug Interactions, highlighting key metrics like accuracy, precision, recall, F1-score, and ROC-AUC. LightGBM and XGBoost outperform other models, indicating their superior predictive capability.

10. BENCHMARKING MODEL PERFORMANCE FOR DDI PREDICTION

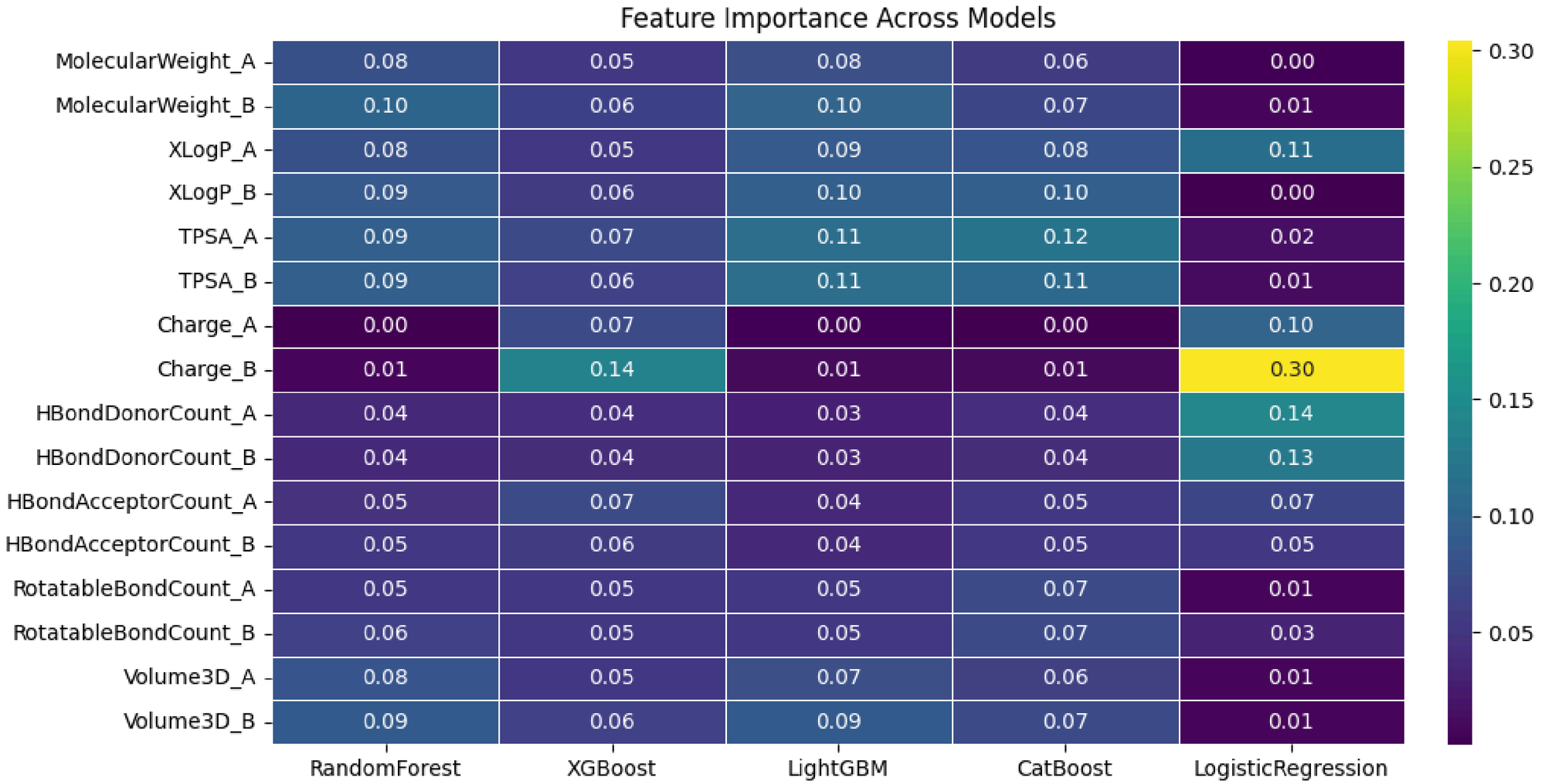


Fig7. Top Performing Model Feature Insight: This chart presents the feature importance ranking for the top-performing model (LightGBM or XGBoost), emphasizing how features like TPSA and XLogP are crucial in determining the likelihood of drug interactions.

10. BENCHMARKING MODEL PERFORMANCE FOR DDI PREDICTION

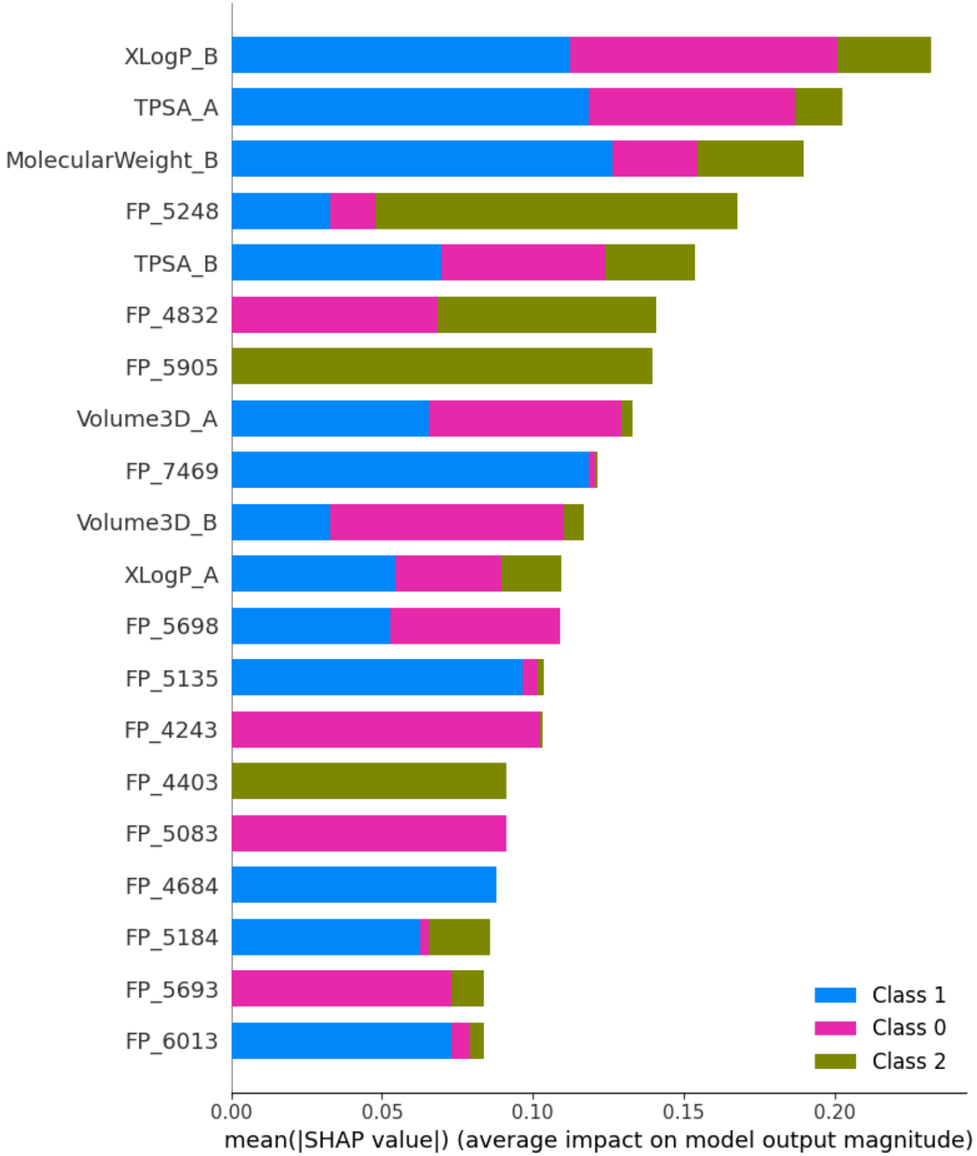


Fig8. Feature Impact on Model Predictions: This visualization represents the mean SHAP values, highlighting the features' average contribution to the model's decision-making process. Key features such as MolecularWeight and TPSA dominate in terms of their influence on predicting drug interactions.

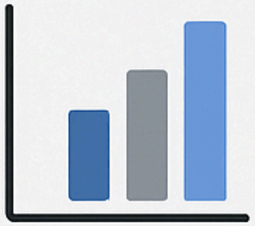
11. RESULTS & KEY INSIGHTS

Drug-Drug Interaction Is Predictable from Molecular Structure

Molecular fingerprints and descriptors alone gave meaninglul predictive signals for GI and metabolic drugs

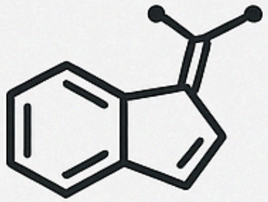
The Outperformed Models

LightGBM demonstrated outstanding performance, delivering highly accurate predictions and showcasing its robust capabilities in the analysis.



Key Features Driving Interactions Identified

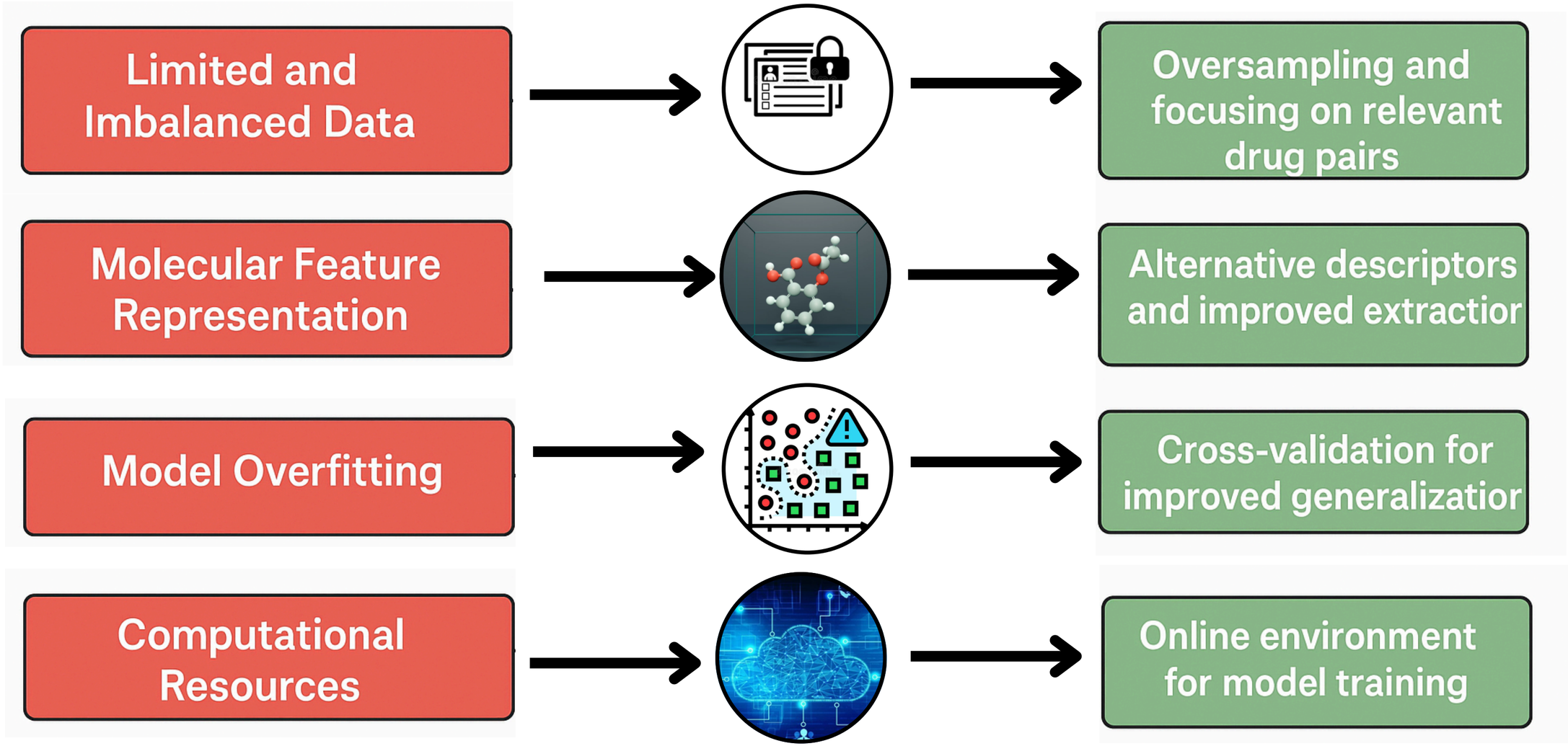
SHAP analysis revealed $XLogP_B$ $TPSA\_A$, MWT_{T_B} and FP428 as top contributors to interaction prediction



Structural Similarity Amplifies DDI Risk

Structurally similar drugs, especially with shared functional groups, showed higher likelihood of interactions

12. OVERCOMING CHALLENGES AND ADDRESSING LIMITATIONS IN DDI PREDICTION



13. ADVANCING HEALTHCARE WITH DDI PREDICTION: APPLICATIONS & IMPACTS

Polypharmacy Management

In clinical settings, especially with elderly patients who are often on multiple medications, your model can be used to predict potential DDIs between prescribed drugs, helping clinicians adjust prescriptions accordingly and avoid negative outcomes.

Pre-clinical Drug Development

Pharmaceutical companies can utilize the model to predict interactions between new drugs and existing ones, ensuring early-stage development focuses on the most promising candidates with minimal risk of harmful interactions.

Personalized Treatment Plans

Your code can be further extended to personalize drug combinations based on a patient's genetic profile, helping create individualized drug regimens that are not only safer but also more effective for specific patient populations.

Drug Safety Monitoring

Post-market surveillance could benefit from this tool, as it could help identify new, rare drug interactions that were not discovered during clinical trials, thus enhancing drug safety in the long term.

14 REFERENCES

1. Xiong, Guoli, Zhijiang Yang, Jiacai Yi, Ningning Wang, Lei Wang, Huimin Zhu, Chengkun Wu et al. "DDInter: an online drug–drug interaction database towards improving clinical decision-making and patient safety." Nucleic acids research 50, no. D1 (2022): D1200-D1207.
2. Davis, Dr E. "Applications of Machine Learning Algorithms in Predicting Drug-Drug Interactions." International Journal of Transcontinental Discoveries, ISSN (2018): 14-19.
3. Wang, Jihong, Xiaodan Wang, and Yuyao Pang. "StructNet-DDI: Molecular Structure Characterization-Based ResNet for Prediction of Drug–Drug Interactions." Molecules 29, no. 20 (2024): 4829.
4. Cheng, Feixiong, and Zhongming Zhao. "Machine learning-based prediction of drug–drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties." Journal of the American Medical Informatics Association 21, no. e2 (2014): e278-e286.
5. PubChem Database. (2023). Chemical information for drug interactions. National Center for Biotechnology Information. Retrieved from PubChem Database. (2023). Chemical Information for Drug Interactions.Retrieved from https://pubchem.ncbi.nlm.nih.gov/
6. Landrum, Greg. Rdkit: Open-source cheminformatics software. 2016.
7. Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

Thank You