MOLECULAR BIOACTIVITY PREDICTION

PRML PROJECT REPORT – PROGRESS REPORT

Sayyam Palrecha - EE22BTECH11047

INTRODUCTION

What is Molecular Bioactivity Prediction?

- Predicts whether a chemical compound will be **biologically active** against a specific target (e.g., protein, receptor).
- Used in **early drug discovery** to filter out non-promising molecules before expensive laboratory experiments.
- Input: molecular structure (SMILES/graph) → Output: **Active / Inactive (binary classification)**

Drug discovery is **slow (10-12 years)** and **expensive.** Only **1 out of ~10,000 compounds** reaches the market.

Predicting bioactivity computationally allows:

- Rapid screening of millions of molecules
- Cost reduction in wet-lab experiments
- Shortened discovery cycle

The table shows: Traditional Approaches (Classical/QSAR Methods)

Method	How it works	Drawbacks
QSAR (Quantitative Structure–Activity Relationship)	Hand-crafted descriptors (physicochemical features, fingerprints) used with ML models	Limited ability to generalize across chemical space
Docking / molecular simulation	Physically places molecule in a target binding site	Computationally expensive; sensitive to conformation and scoring bias
Rule-based filters (Lipinski, PAINS)	Hard-coded heuristics to eliminate bad molecules	Not predictive; only filters candidates

MOTIVATION/CONTEXT

We require a rapid, data-driven method to predict whether a compound is active or inactive before experimental validation.

Why Machine Learning (ML)?

Benefit	What It Solves in Drug Discovery
Learns relationships between molecular structure and bioactivity	Eliminates manual feature engineering (rules/QSAR)
Fast inference (milliseconds per molecule)	Enables **virtual screening of millions of molecules**
Works with small or moderate datasets	Ideal when experimental data is limited

- ML accelerates drug discovery by predicting molecular bioactivity without costly experiments.
- GNNs go one step further — letting the model “see” the molecule as a **graph of atoms and bonds**, not just numbers.

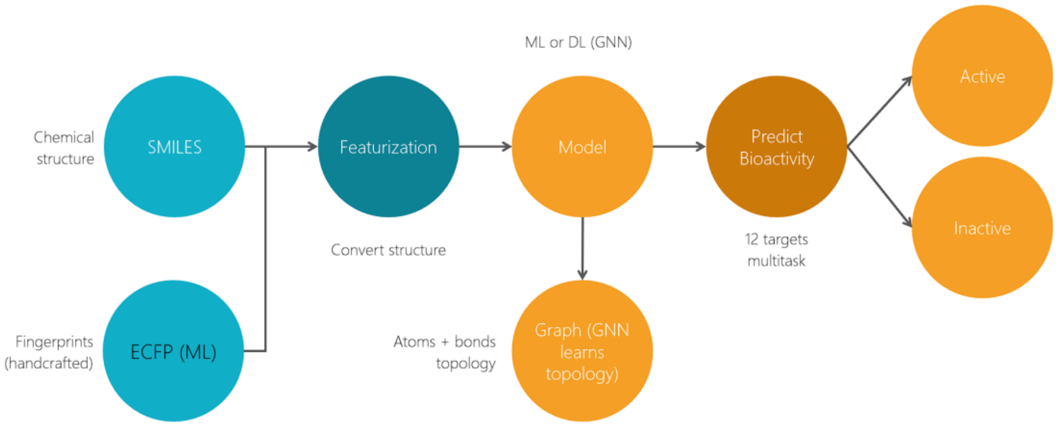
Why These Models?

Model	Why used in our project?	Strength
SVM	Strong baseline classifier for small datasets	Maximizes margin between Active / Inactive classes
Random Forest (RF)	Handles noisy and imbalanced chemical data	Robust, interpretable (feature importance)
XGBoost	Often state-of-the-art in tabular molecular data	Learns complex/non-linear feature interactions; efficient
Graph Neural Networks (GNN)	Learns directly from **molecular graphs** instead of fingerprints	Captures chemical structure + topology automatically

PROBLEM STATEMENT & PROJECT GOALS

Given molecular structures (SMILES), build a model that predicts bioactivity across multiple biological targets (Tox21 - 12 tasks), despite label imbalance and noisy data.

Challenge	Meaning
Multi-task classification	One molecule → 12 outputs (bioactivity across targets)
Severe class imbalance	Very few Active vs many Inactive compounds
Feature representation problem	Fingerprints lose topological structure of molecules



Specific Goals

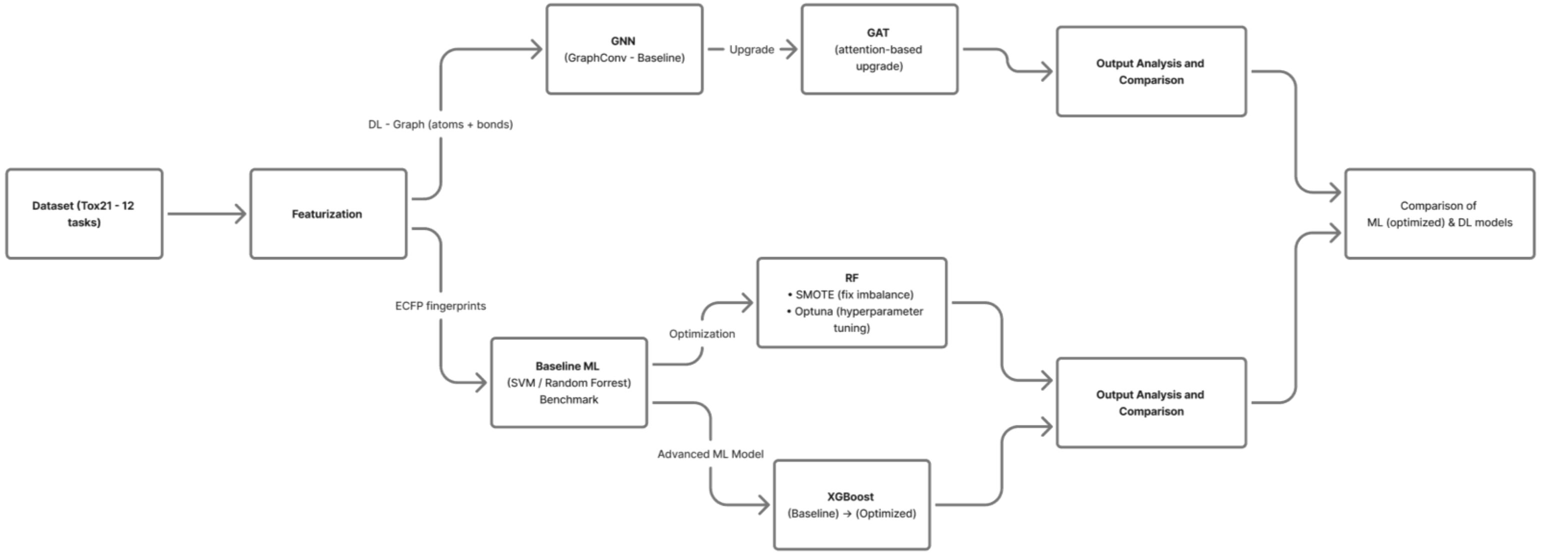
- Train baseline ML models on fingerprint representations and optimize using imbalance handling + hyperparameter tuning
- Advance to Deep Learning (GNNs) to learn directly from molecular graphs
- Compare ML vs GNN performance and interpret learned molecular features
- Develop an **SOTA-inspired hybrid modelling approach** to overcome ML/DL limitations

APPROACH (MATERIALS & METHODS)

Dataset Used: **Tox21 (Toxicology in the 21st Century)** - public benchmark for predictive toxicology.

- **12 binary toxicity endpoints** (Nuclear Receptors + Stress Response pathways)
- **~8,000 unique compounds** (SMILES / SDF) with a **multi-task, multi-label** setup
- **Strong class imbalance** (far fewer actives than inactives)

Approach: From Baseline → Optimization → Deep Learning (flowchart)


Dataset (Tox21 - 12 tasks)
Featurization
DL - Graph (atoms + bonds)
GNN (GraphConv - Baseline)
Upgrade
GAT (attention-based upgrade)
Output Analysis and Comparison
ECFP fingerprints
Baseline ML (SVM / Random Forrest) Benchmark
Optimization
RF
• SMOTE (fix imbalance)
• Optuna (hyperparameter tuning)
Advanced ML Model
XGBoost (Baseline) → (Optimized)
Output Analysis and Comparison
Comparison of ML (optimized) & DL models

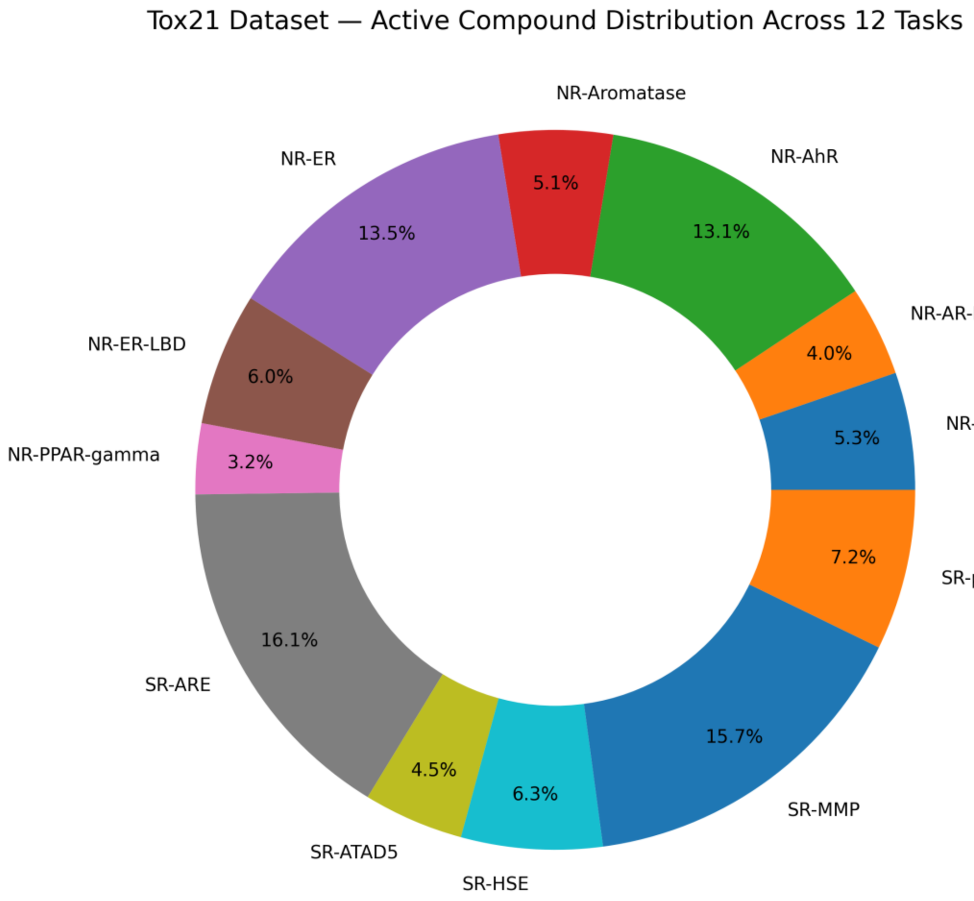
APPROACH (MATERIALS & METHODS)

Metrics Used:

- **ROC-AUC**: How well the model separates Active vs Inactive compounds
- **PR-AUC**: Performance under severe class imbalance
- **F1 Score**: Balance between precision & recall
- **Precision/Recall**: False positives vs false negatives trade-off

How results are analysed:

- Compare metrics across models:
  - Baseline ML → Optimized ML → GNN
- Visual comparisons:
  - Bar plots for ROC-AUC/PR-AUC
  - Training time comparison



IMPLEMENTATION (BASELINE: SVM & RANDOM FOREST)

Support Vector Machine: Find an optimal decision boundary that separates **Active** vs **Inactive** molecules using molecular fingerprints (ECFP vector)

Each molecule is represented as:

$$x \in \{0,1\}^d \text{ (ECFP fingerprint)}$$

SVM computes a nonlinear decision function:

$$K\left(x_i, x_j\right) = e^{-\gamma \|x_i - x_j\|^2} \quad (rbf\ kernel)$$

The classifier solves:

$$\min \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \varepsilon_i$$

$$y_i(\boldsymbol{w}^T \phi(x_i) + b) \geq 1 - \varepsilon_i$$

Outcome: For each molecule x, SVM outputs:

$$\hat{y} = sgn\left(\sum_i \alpha_i y_i K(x_i, x)\right)$$

Where, ε_i are slack variables for incorrect classifications

Random Forest: Learn decision rules from fingerprint bits to classify bioactivity.

Each tree partitions molecular space based on fingerprint features:

$$split: x_j \leq t$$

Forest prediction:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$$

Where $h_t(x)$ prediction from the tree t, T is number of trees

RF probability output (for Active class):

$$P(Active|x) = \frac{1}{T} 1_{\{h_t(x) = Active\}}$$

What outputs from these models tell us:

- SVM gives **decision boundary** → helps understand separability limits of descriptor space.
- RF gives **feature importance rankings** → used later for interpretability & optimization.

IMPLEMENTATION (OPTIMIZED RF & XGBOOST)

Handling Class Imbalance using **SMOTE**

SMOTE synthesizes new minority (Active) samples in the feature space:

$$x_{new} = x_i + \lambda\left(x_i^{(NN)} - x_i\right), \lambda \sim U(0,1)$$

Where, x_i: minority (Active) sample,

$x_i^{(NN)}$: nearest neighbor in minority class

RF - Hyperparameter Optimization (Optuna search)

$$\min_{\theta} \mathcal{L}_{val}\left(f_\theta(X)\right)$$

Where $\theta = \{RF\ parameters: number\ of\ trees, tree\ depth, \dots\}$

The objective returns:

$$\theta^* = \arg\max_{\theta} PR - AUC_{validation}$$

Since Tox21 is severely imbalanced, we optimize using **PR-AUC** rather than accuracy.

XGBoost (Gradient Boosted Trees)

Learns trees sequentially, minimizing the loss at each boosting round.

Model prediction:

$$\hat{y} = \sum_{k=1}^{K} f_k(x), f_k \in \mathcal{F}$$

Training minimizes:

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$

Where, $l(y_i, \hat{y}_i)$: loss (logistic loss for classification)

$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2$ (model complexity penalty)

T: number of leaves, ω: leaf weights

Leaf-wise update rule (core of XGBoost)

$$\omega^* = -\frac{\sum_i g_i}{\sum_i h_i + \lambda}$$

Where, g_i: first derivative (gradient) of loss

h_i: second derivative (Hessian)

IMPLEMENTATION (DL: GRAPHCONV → GAT)

Molecules as Graphs: Every molecule is converted into a graph

$$G = (V, E)$$

GraphConv (Message Passing Neural Network)

Message passing updates node (atom) embeddings layer by layer:

$$h_v^{(k+1)} = \sigma\left(W^{(k)} \cdot \sum_{u \in \mathcal{N}(v)} h_u^{(k)} + b^{(k)}\right)$$

Where, $\mathcal{N}(v)$: neighbors of atom v(bonded atoms)

$h_u^{(k)}$: feature of neighbor atom u at layer k

$W^{(k)}$, $b^{(k)}$: learnable weights, σ: activation (ReLU)

After multiple layers and pooling:

$$H = \left[h_1^{(K)}, h_2^{(K)}, \dots, h_n^{(K)}\right]$$
$$z = pooling(H)$$
$$\hat{y} = \sigma(Wz + b)$$

GAT (Graph Attention Network): Instead of treating all neighbors equally, GAT **assigns learned attention weights** to neighbors

$$e_{uv} = a(Wh_u, Wh_v)$$

Attention coefficients:

$$\alpha_{uv} = \frac{e^{e_{uv}}}{\sum_{k \in \mathcal{N}(v)} e^{e_{vk}}}$$

Node update:

$$h_v^{(k+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{uv} \cdot W h_u^{(k)}\right)$$

Important: α_{uv} tells us **which atoms matter** for bioactivity. This introduces **interpretability** into the model.

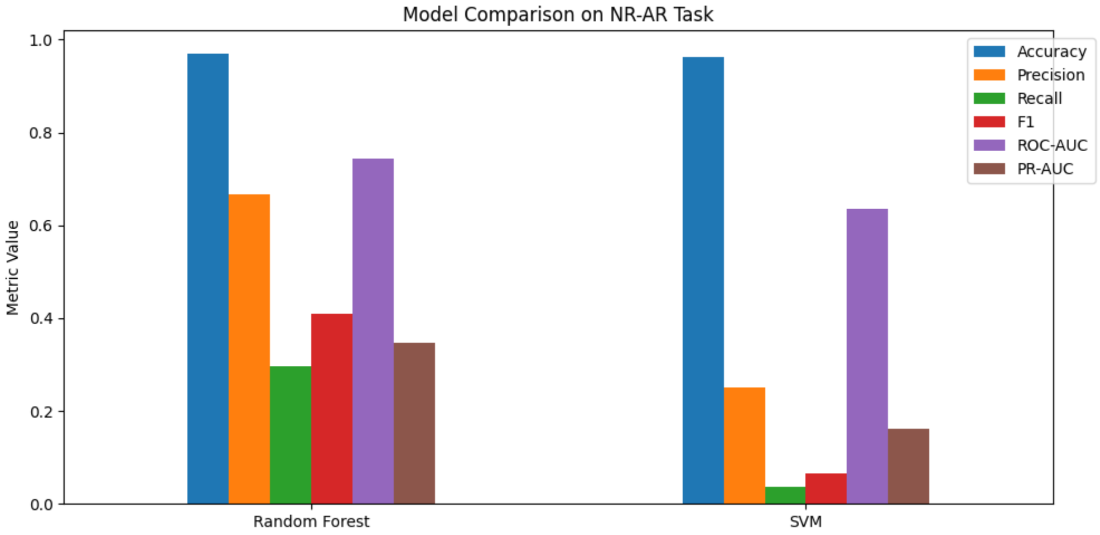
Output of this stage

Model	Output
GraphConv	Learns structural (topological) patterns
GAT	Learns which atoms contribute most to bioactivity

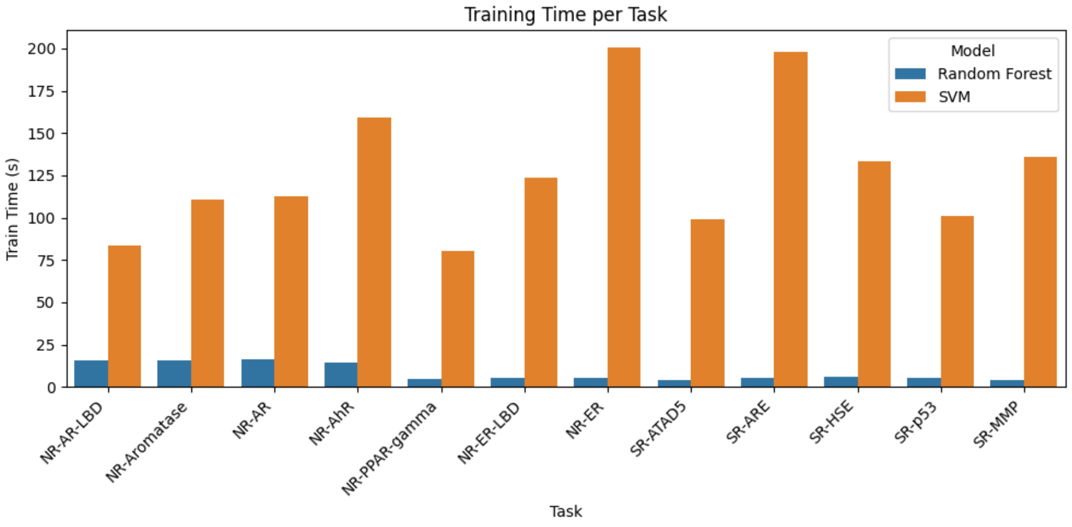
GraphConv learns structure. GAT learns structure + importance.

INTERMEDIATE RESULTS

For **baseline ML models**, RF and SVM (model comparison done for a single task)



Train time comparison per task (RF vs SVM)

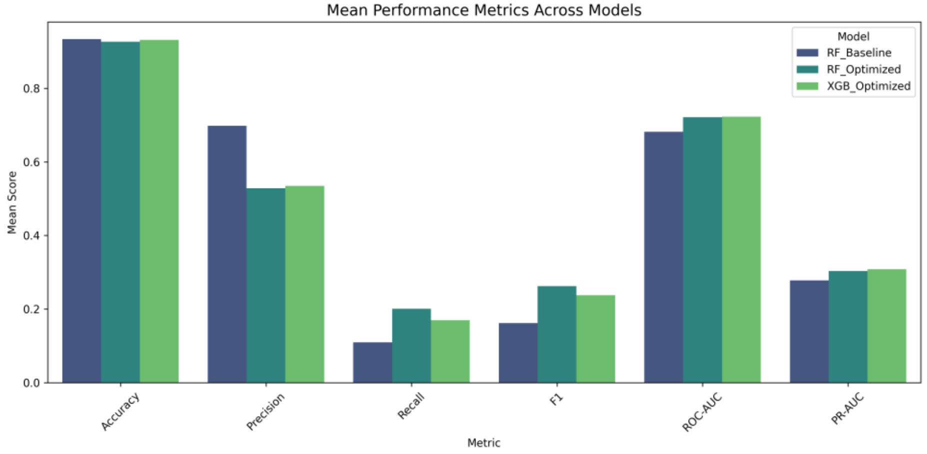


Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	PR-AUC	Train Time (s)
Random Forest	0.9298	0.4333	0.0886	0.1376	0.6895	0.2324	8.50
SVM	0.9266	0.3361	0.0702	0.1107	0.6769	0.2019	128.16

INTERMEDIATE RESULTS

Optimized RF and XGBoost - mean score comparison

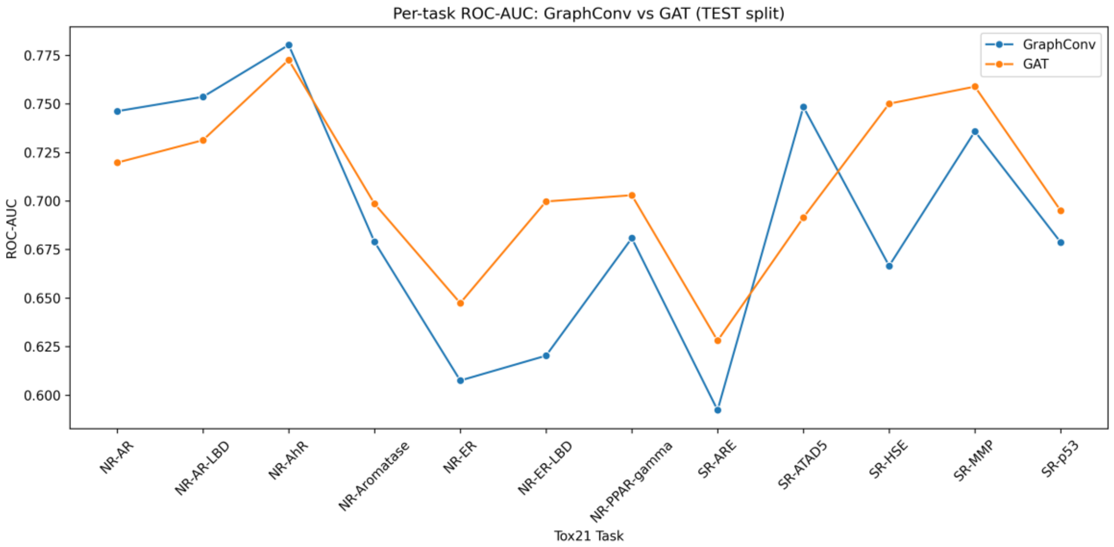
Across Baseline and Optimized Models (for four tasks)



Improvement Over Baseline (Mean Across 4 Tasks)

Metric	RF Improvement	XGB Improvement
Accuracy	**−0.75%**	**−0.20%**
F1 Score	**+62.05%**	**+46.49%**
ROC-AUC	**+5.87%**	**+6.03%**
PR-AUC	**+9.16%**	**+10.95%**

GNN Results (Across all tasks):



Mean scores (Across all tasks):

Model	ROC-AUC (mean)	PR-AUC (mean)
GraphConv	**0.7154**	**0.2517**
GAT	**0.7302**	**0.2185**

KEY LEARNING & CHALLENGES SOLVED

Challenges and learning

- Bioactivity dataset was **highly imbalanced** (very few Active molecules). Solved using **SMOTE oversampling** + **cost-sensitive learning**.
- Baseline ML (RF/SVM) gave **high accuracy but low recall**, meaning actives were missed. Hyperparameter search (Optuna) improved **Recall, F1, ROC-AUC and PR-AUC**.
- XGBoost outperformed RF in **PR-AUC and ROC-AUC**, confirming effectiveness on sparse chemical fingerprints.
- GraphConv/GAT performed better in **generalization across tasks**. **GAT gives the highest ROC-AUC value**.
- Observed that **the attention mechanism (GAT) improves the** learning of structural molecular features.

Key takeaways

- Optimizing ML models required **balancing recall vs precision**.
- GNNs demonstrated **better chemical structure awareness** than ECFP+ML models.
- ML is fast & interpretable; GNN is powerful & scalable.

FUTURE STEPS AND REFERENCES

Future Steps

- **Explore a hybrid ML + DL framework:** Combine the strengths of classical ML models with GNNs to build a meta-model that utilises fingerprints + learned graph embeddings.
- Investigate model-level optimization
    - Evaluate techniques like **feature selection + dimensionality reduction** before SVM/ML models.
    - Experiment with **GNN hyperparameter tuning** (GIN depth, attention heads, dropout, learning rate schedules) to improve performance & reduce training time.

References

- “Toxicology in the 21st Century (Tox21)” - Dataset description (NIH/NCATS public documentation)
- Chawla, N. V., et al. (2002). *SMOTE: Synthetic Minority Over-sampling Technique.*
- Optuna Documentation — https://optuna.org/
- Kipf, T. N., & Welling, M. (2017). *Semi-Supervised Classification with Graph Convolutional Networks.*
- Veličković, P. et al. (2018). *Graph Attention Networks.*