MACHINE LEARNING / AI PROJECTS

NIKHIL DHIMAN

August 14, 2025

PREDICTIVE MAINTENANCE: PROACTIVE SAFETY IN INDUSTRIAL OPERATIONS

At Nutrien, safety means Everyone Home Safe, Every Day. Predictive Maintenance saves lifes

Why it matters:

- **SIF Prevention: Early hazard detection stops major incidents.**
- **Environmental Protection: Prevents leaks, emissions, and hazardous releases.**

Real-World Incidents
West Fertilizer (2013) – Ammonium nitrate blast; 15 dead, 200+ injured
Williams Olefins (2013) – Heat exchanger rupture; 2 dead, 167 injured
Clairton Coke Works (2025) – Gas explosion; 2 dead, 10+ injured

Code: https://github.com/NikhilDhiman/Artwork-Mapped-Using-ML

EXPLORATORY DATA ANALYSIS (EDA)

Data Source: https://archive.ics.uci.edu/ml/machine-learning-databases/00601/ai4i2020.csv

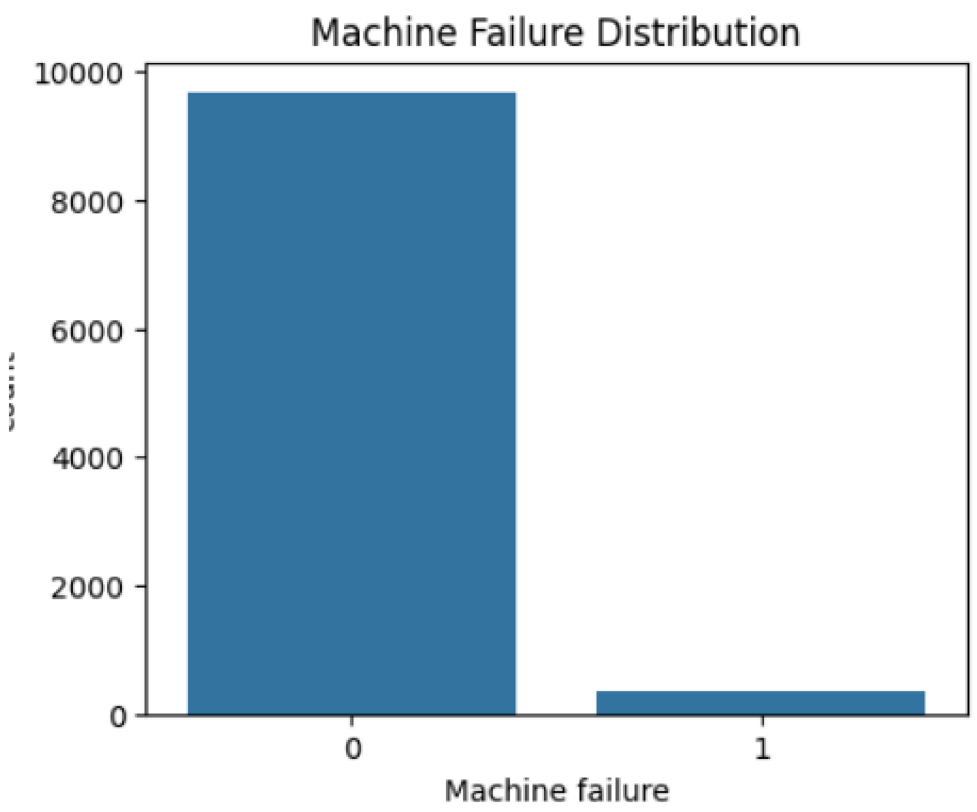
Shape: (10000, 14)

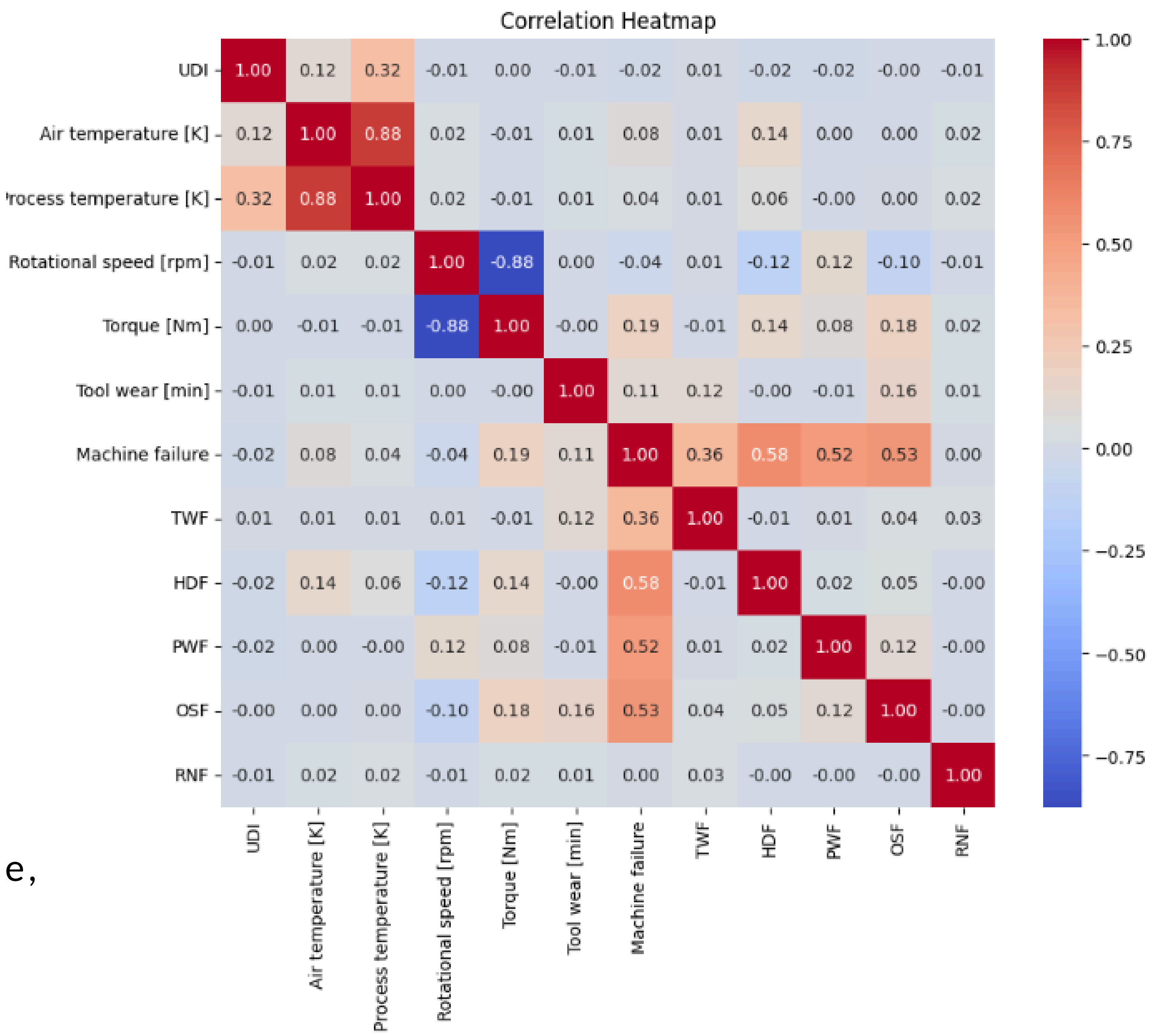
	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	TWF	HDF	PWF	OSF	RNF
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	0	0	0	0	0

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   UDI                      10000 non-null  int64
 1   Product ID               10000 non-null  object
 2   Type                     10000 non-null  object
 3   Air temperature [K]      10000 non-null  float64
 4   Process temperature [K]  10000 non-null  float64
 5   Rotational speed [rpm]   10000 non-null  int64
 6   Torque [Nm]              10000 non-null  float64
 7   Tool wear [min]          10000 non-null  int64
 8   Machine failure          10000 non-null  int64
 9   TWF                      10000 non-null  int64
 10  HDF                      10000 non-null  int64
 11  PWF                      10000 non-null  int64
 12  OSF                      10000 non-null  int64
 13  RNF                      10000 non-null  int64
dtypes: float64(3), int64(9), object(2)
memory usage: 1.1+ MB
None
```

```
Missing Values per Column:
UDI                        0
Product ID                 0
Type                       0
Air temperature [K]        0
Process temperature [K]    0
Rotational speed [rpm]     0
Torque [Nm]                0
Tool wear [min]            0
Machine failure            0
TWF                        0
HDF                        0
PWF                        0
OSF                        0
RNF                        0
dtype: int64
```

FAILURE DISTRIBUTION





How I handled class imbalance:
Class-weighted models: Balanced weights in Logistic Regression, Decision Tree, Random Forest.
XGBoost weighting: scale_pos_weight = #neg / #pos for minority class focus.
Stratified CV: Preserve class ratios in all folds.
Metrics: Used ROC-AUC & PR-AUC (PR-AUC for imbalance).

MODEL CHOICES

Model Choices & Rationale

- **Logistic Regression** – Simple, interpretable baseline; fast; class_weight='balanced' for imbalance.
- **Decision Tree** – Captures non-linear rules; handles mixed data; visualizable; balanced weights.
- **Random Forest** – Ensemble of trees; reduces overfitting; feature importance; balanced weights.
- **XGBoost** – High-performance boosting; handles complex patterns; scale_pos_weight for imbalance; tuned for best results.

XGBoost Fine-Tuning
Pipeline: Preprocessor + XGBClassifier.
Search: RandomizedSearchCV (80 configs, F1 score).
CV: Stratified 5-fold.

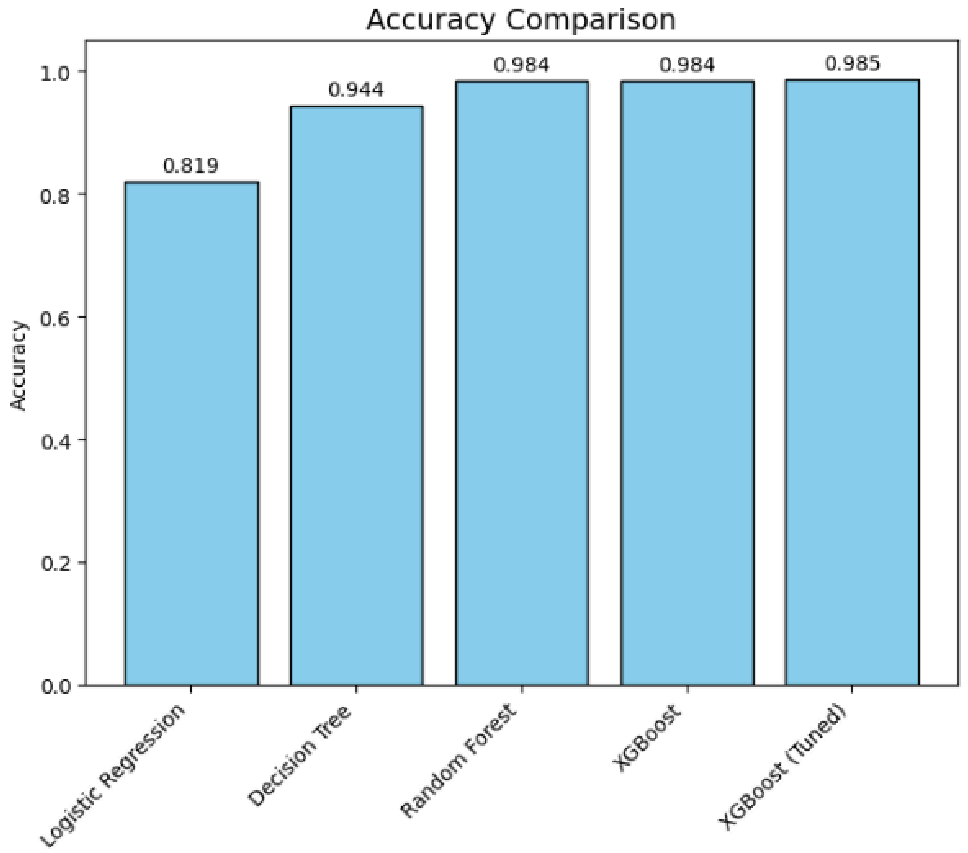
Key Params Tuned:
n_estimators, learning_rate, max_depth,
min_child_weight, subsample, colsample_bytree,
reg_alpha, reg_lambda.

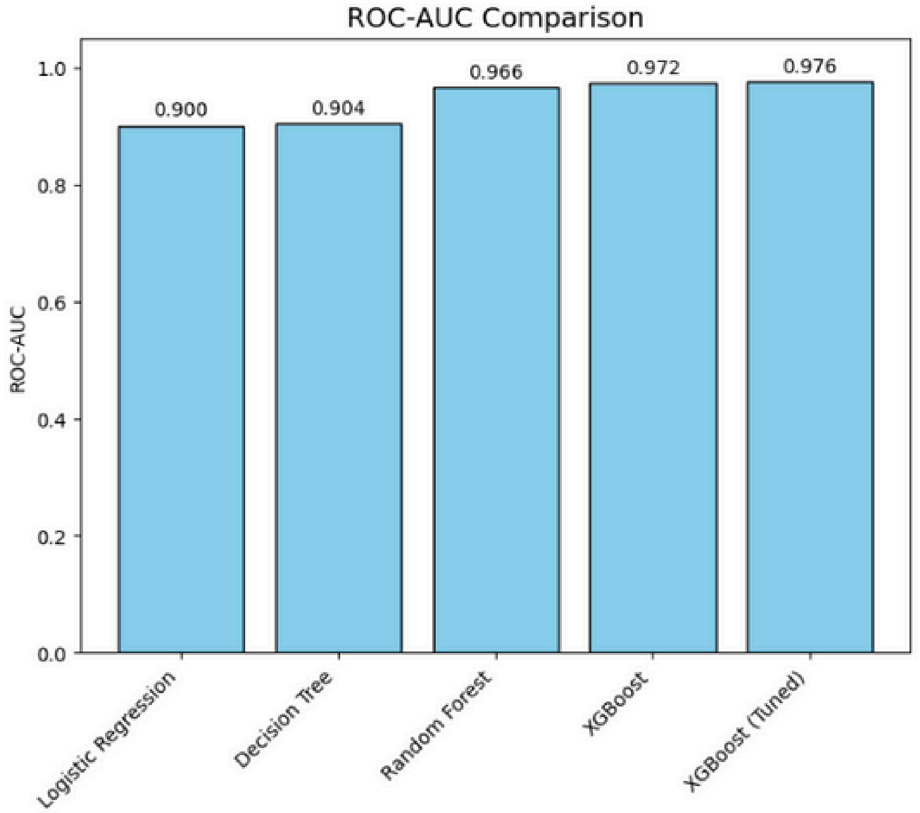
RESULTS

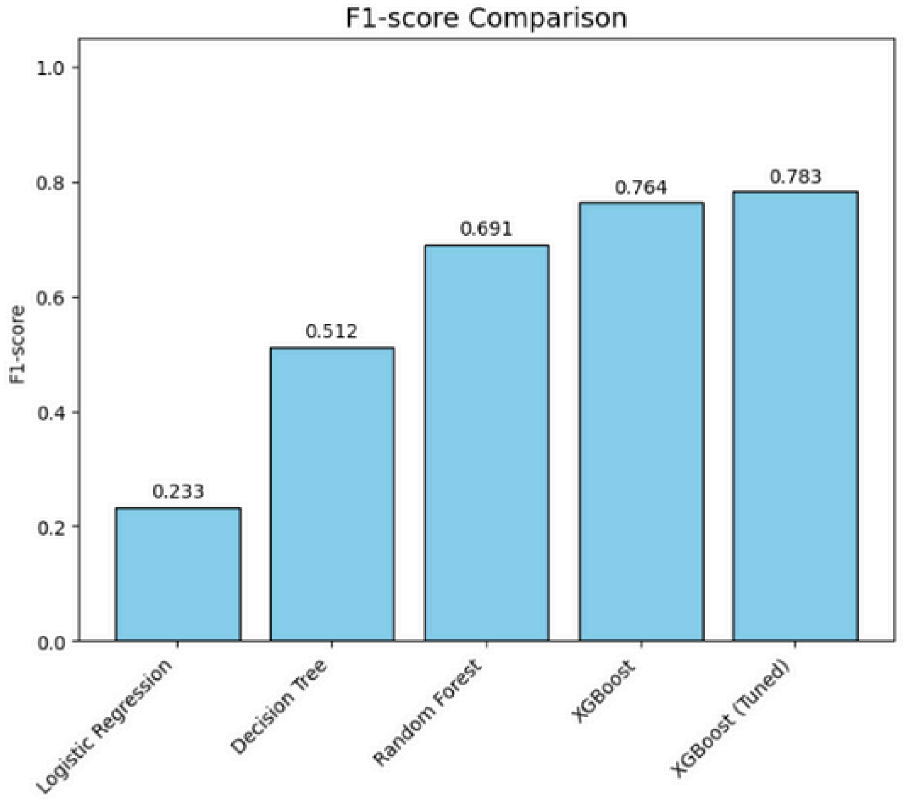
Metrics:
Accuracy: How often the model is right (can be misleading if one class is much bigger).
Precision: When the model says “positive,” how often it’s correct.
Recall: Of all the real positives, how many the model finds.
F1-score: A balance between precision and recall.
ROC-AUC: How well the model tells the two classes apart.
PR-AUC: Focuses on how well the model finds the important (positive) cases in imbalanced data.

	Model	Accuracy	Precision	Recall	F1-score	ROC-AUC	PR-AUC
0	Logistic Regression	0.8194	0.136120	0.808253	0.232938	0.900340	0.426264
1	Decision Tree	0.9435	0.362002	0.873222	0.511771	0.904406	0.731281
2	Random Forest	0.9836	0.949822	0.548903	0.690572	0.966190	0.818113
3	XGBoost	0.9843	0.783645	0.749342	0.763835	0.972261	0.829322
4	XGBoost (Tuned)	0.9853	0.784352	0.784723	0.783279	0.976223	0.842558

RESULTS


Accuracy Comparison
Accuracy
0.819
0.944
0.984
0.984
0.985
Logistic Regression
Decision Tree
Random Forest
XGBoost
XGBoost (Tuned)


ROC-AUC Comparison
ROC-AUC
0.900
0.904
0.966
0.972
0.976
Logistic Regression
Decision Tree
Random Forest
XGBoost
XGBoost (Tuned)


F1-score Comparison
F1-score
0.233
0.512
0.691
0.764
0.783
Logistic Regression
Decision Tree
Random Forest
XGBoost
XGBoost (Tuned)

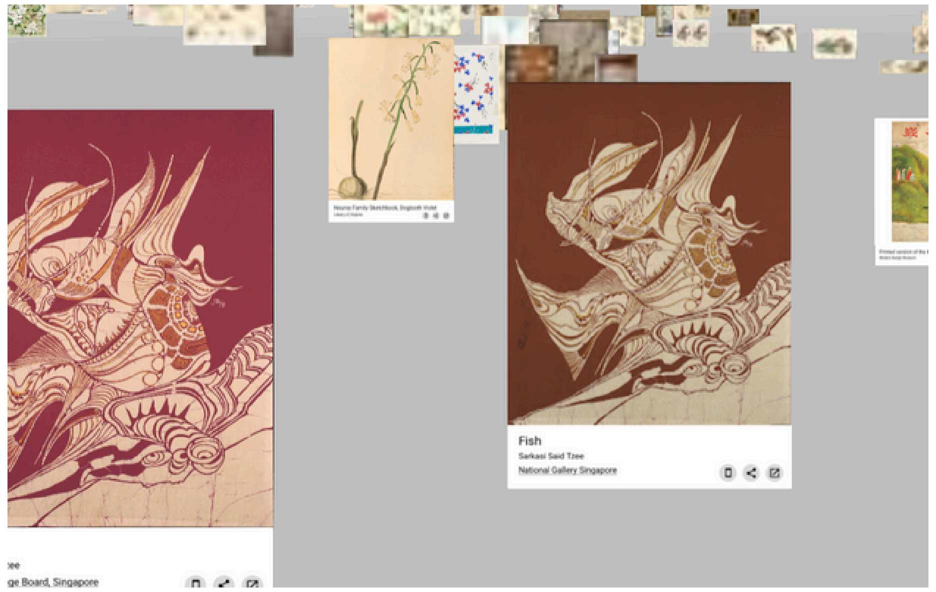
ARTWORK MAPPED USING ML

An interactive 3D visualization of 120K artworks mapped by visual similarity using ML and dimensionality reduction.

UNSUPERVISED LEARNING

FEATURE EXTRACTION

DIMENSIONALITY REDUCTION AND CLUSTERING



Live Demo: https://3d-umap-cs5660.vercel.app
Code: https://github.com/NikhilDhiman/Artwork-Mapped-Using-ML

FLOW OF PROJECT

PRE-DATA CHECKS

- Using OpenCV
- Duplicate and corrupt image filtering
- Blurry Image Detection

EXTRACT HIGH-DIMENSIONAL FEATURE

- TensorFlow Dataset Pipeline
- ResNet50 Model
- HDF5 Feature Storage

DIMENSIONALITY REDUCTION

- PCA
- UMAP

BUILD AN INTERACTIVE 3D LANDSCAPE WHERE

- HDBSCAN
- Three JS
- HTML
- CSS

PRE-DATA CHECKS

```
# Total Valid Images

# Define valid image file extensions
valid_exts = ('.jpg', '.jpeg', '.png')
# Recursively walk through IMAGES_DIR and collect paths to all valid image files
all_image_paths = [
    os.path.join(root, f)
    for root, _, files in os.walk(IMAGES_DIR)
    for f in files
    if f.lower().endswith(valid_exts) and not f.startswith(".")
]
# Print the total number of images found
print(f"Total images found: {len(all_image_paths)}")
```

```
Total images found: 111668
```

Checking for duplicates: 100% Completed
No Duplicate Found

Checking for blurriness: 100% Completed
No Blurriness Found

EXTRACT HIGH-DIMENSIONAL FEATURE

Goal: Turn each artwork into a 2048-number vector capturing its style & content.

Feature Extraction Pipeline

- Preprocessing: Resize to 224×224, normalize, clean corrupt images.
- Model: **ResNet50** (ImageNet pretrained, Global Avg Pooling → 2048-D/image).
- Batch Processing: **TF Dataset AP**I, **batch=32**, **parallel load & prefetch**.
- Storage: Features in HDF5, filenames in NumPy, resume support (111,668 images).

Why ResNet50?

Sees fine details, captures patterns, and gives a fixed-size summary.

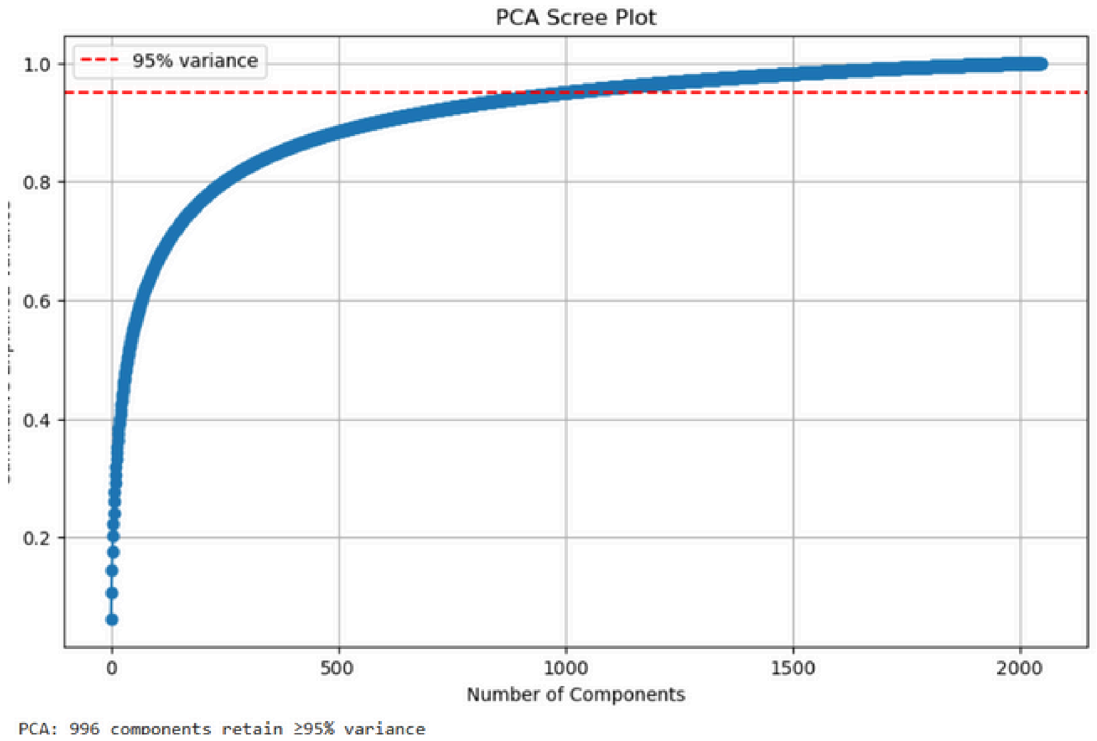
Why Pretrained CNN?

Already trained on millions of images: fast, accurate, works without labels.

DIMENSIONALITY REDUCTION

We used PCA before UMAP to compress the 2048-dimensional features down to only the components that explain most of the variance (≥95%), because:

- **Speeds up UMAP**
- **Reduces noise**

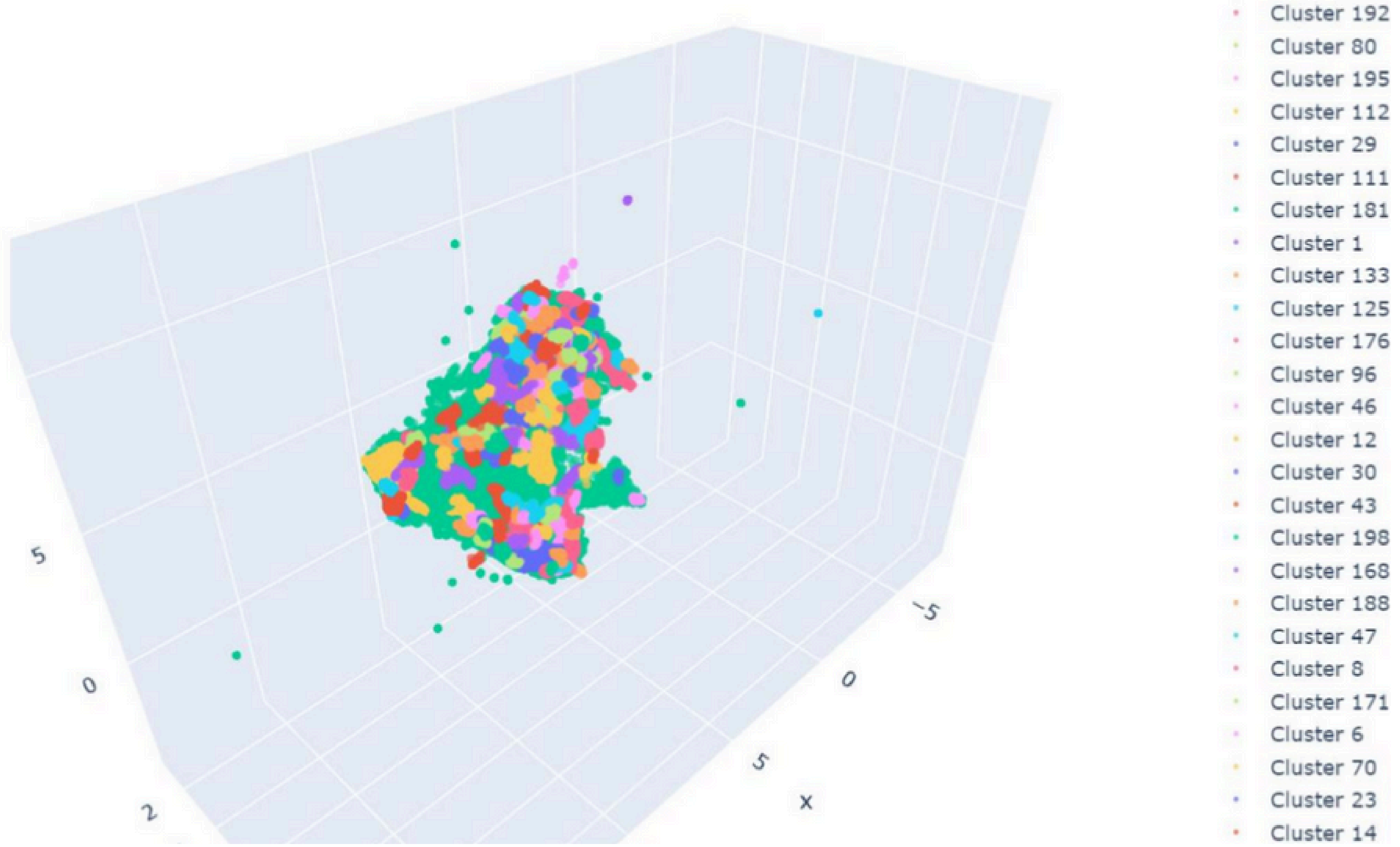


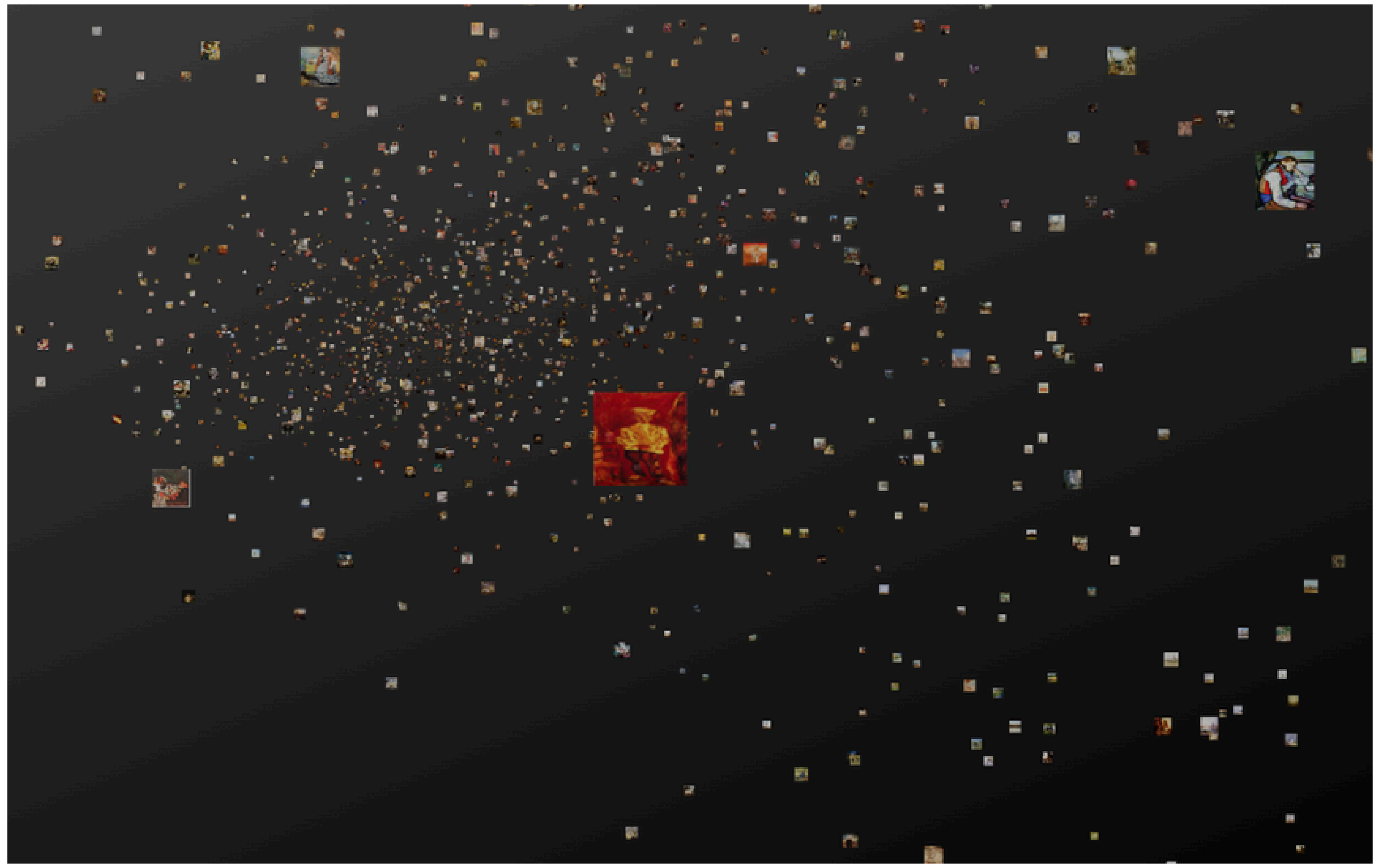
So, PCA acts as a denoising + dimensionality reduction pre-step before the more flexible, nonlinear UMAP mapping.

UMAP + HDBSCAN

Reduce features to 3D, then cluster to find natural groupings

- **Evaluate**: Silhouette score for cluster quality
- **Optimize**: Grid search best UMAP parameters
- **Visualize**: Plotly 3D scatter:





Live Demo: https://3d-umap-cs5660.vercel.app

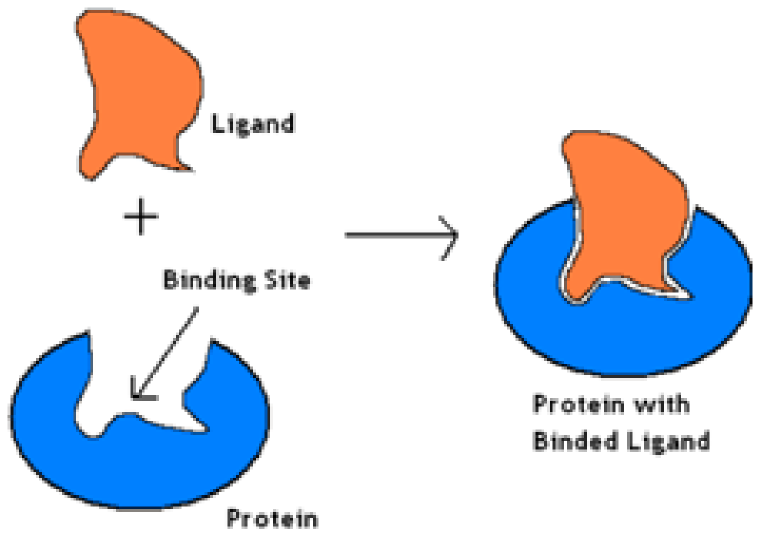
PHYSICS-GUIDED DEEP GENERATIVE MODEL FOR NEW LIGAND DISCOVERY

Generative AI Model to generate new medicine molecules

SEMI-SUPERVISED

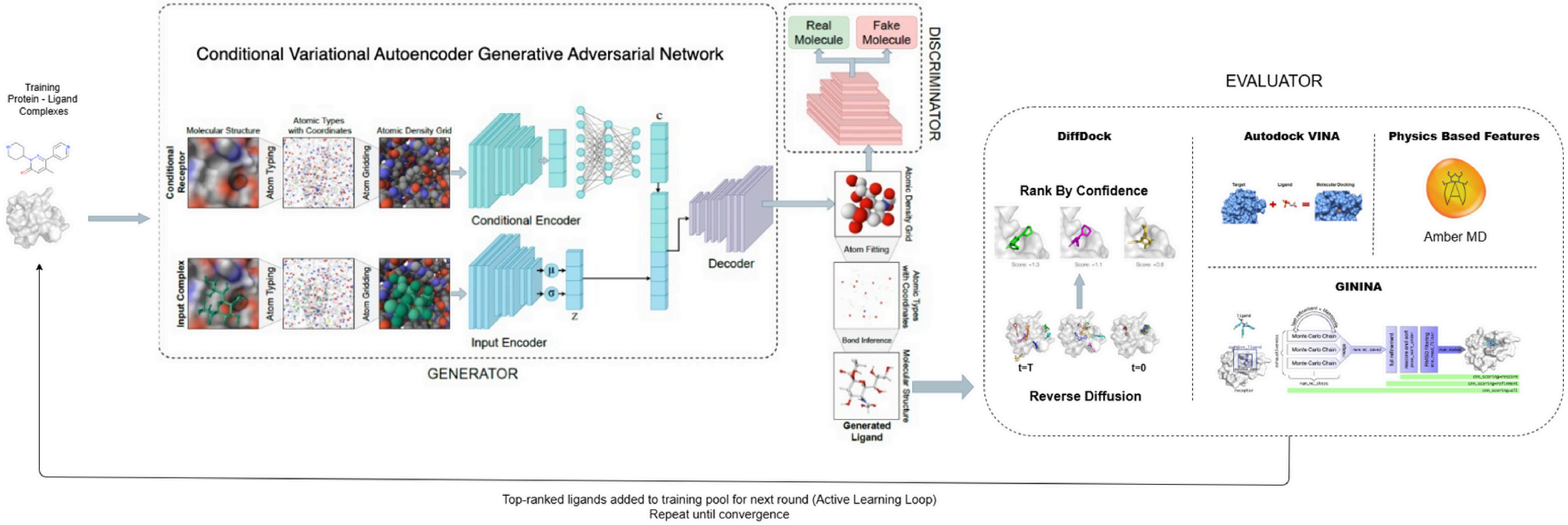
GENERATIVE AI - DEEP LEARNING

ACTIVE LEARNING



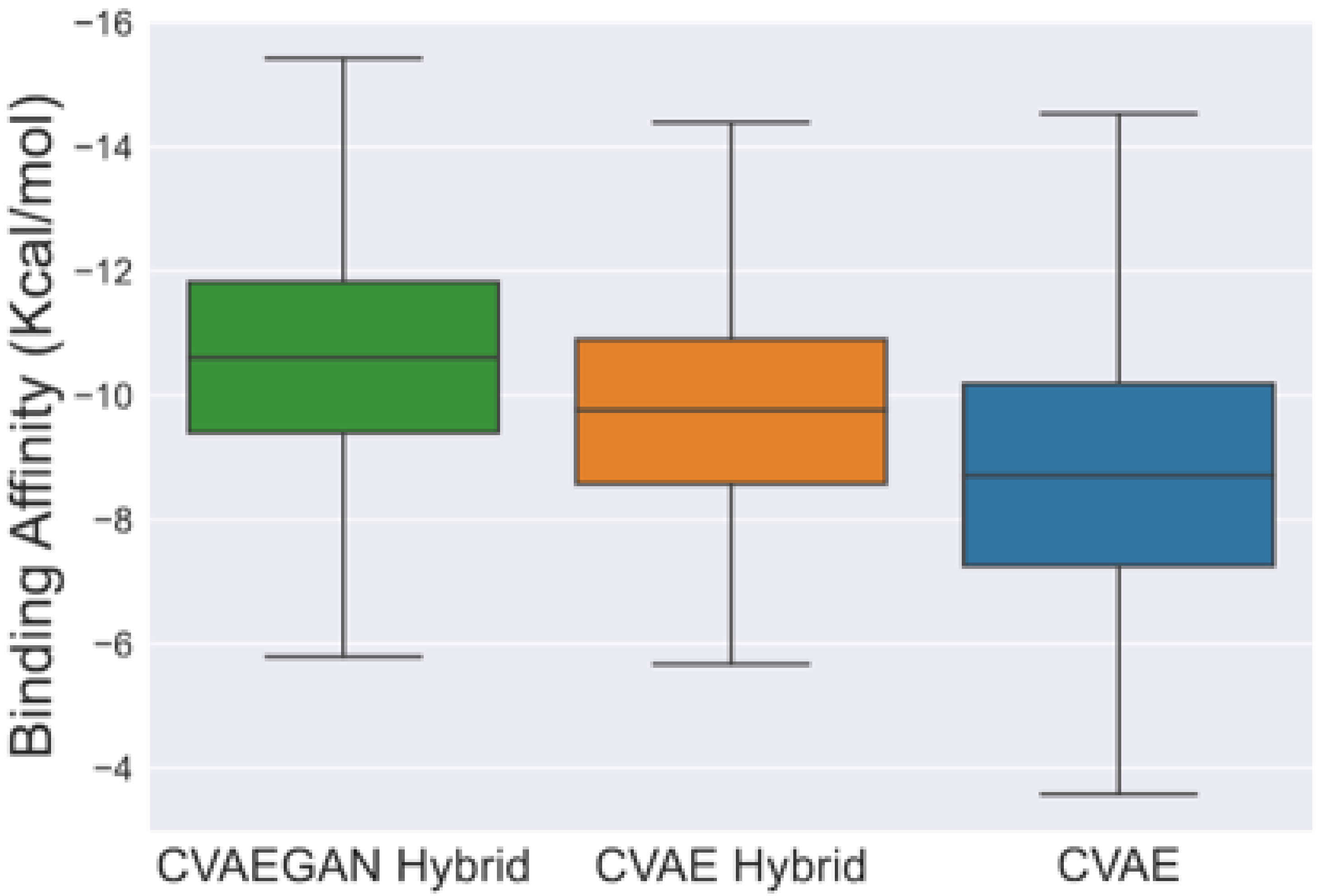
Publication Link: https://www.cell.com/biophysj/abstract/S0006-3495(24)02507-4

ARCHITECTURE



CVAEGAN framework for active learning. Ligands are generated using the CVAEGAN model, evaluated through docking, binding free energy, and physics-based metrics, and top candidates are iteratively fed back for model retraining.

RESULTS



Binding Affinity (Kcal/mol)
−16
−14
−12
−10
−8
−6
−4
CVAEGAN Hybrid
CVAE Hybrid
CVAE

OTHER ML/AI DATA ANALYSIS PROEJECTS

- **Early Skin Cancer Detection: Bringing Dermatology to Everyone**
  https://github.com/NikhilDhiman/Early-Skin-Cancer-Detection-Bringing-Dermatology-to-Everyone

- **Amazon Employee Access Challenge**
  https://github.com/NikhilDhiman/Amazon-Employee-Access-Challenge

- **KNN Classification using Scikit learn**
  https://github.com/NikhilDhiman/KNN-Classification-using-Scikit-learn

- **IMDb Movie Data Analysis**
  https://github.com/NikhilDhiman/IMDb-Movie-Data-Analysis

Github: https://github.com/NikhilDhiman
Portfolio Website: https://nikhildhiman.me/

THANK YOU

Nutrien Ag Solutions®

for the time seeing my projects