ChemML: A Machine Learning And Informatics Program Package For The Analysis, Mining, And Modeling Of Chemical And Materials Data

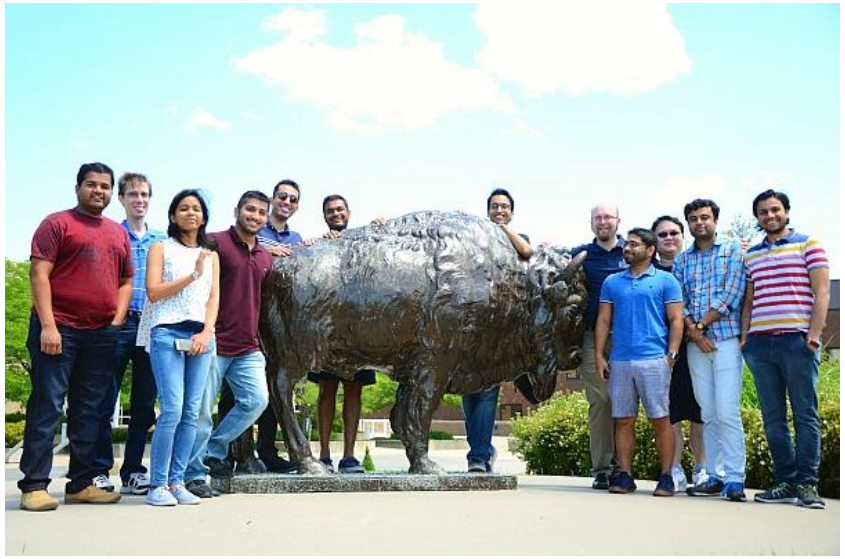
Nitin Murthy, Johannes Hachmann

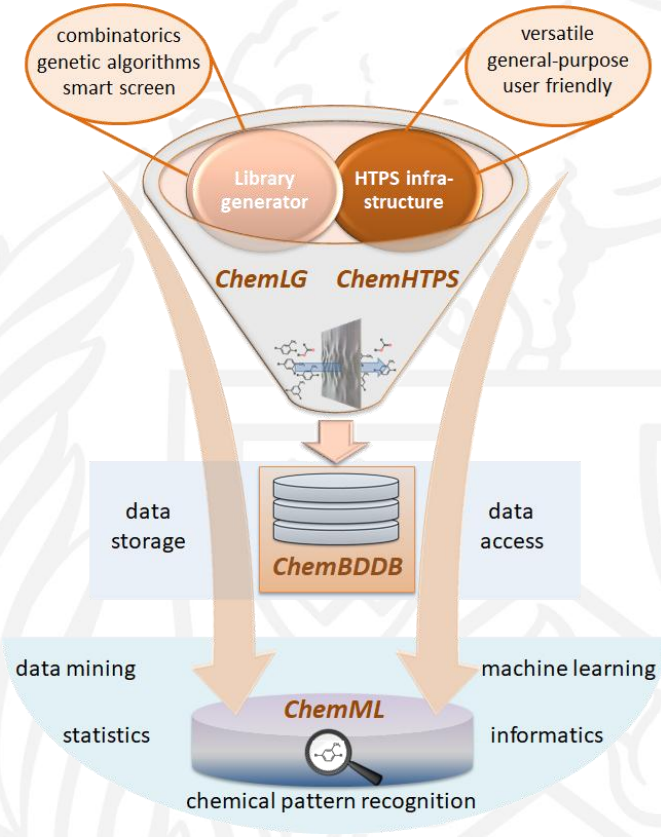
nitinmad@buffalo.edu

hachmann@buffalo.edu

University at Buffalo
Department of Chemical and Biological Engineering
School of Engineering and Applied Sciences

Hachmann Lab And The *ChemEco* Software Ecosystem

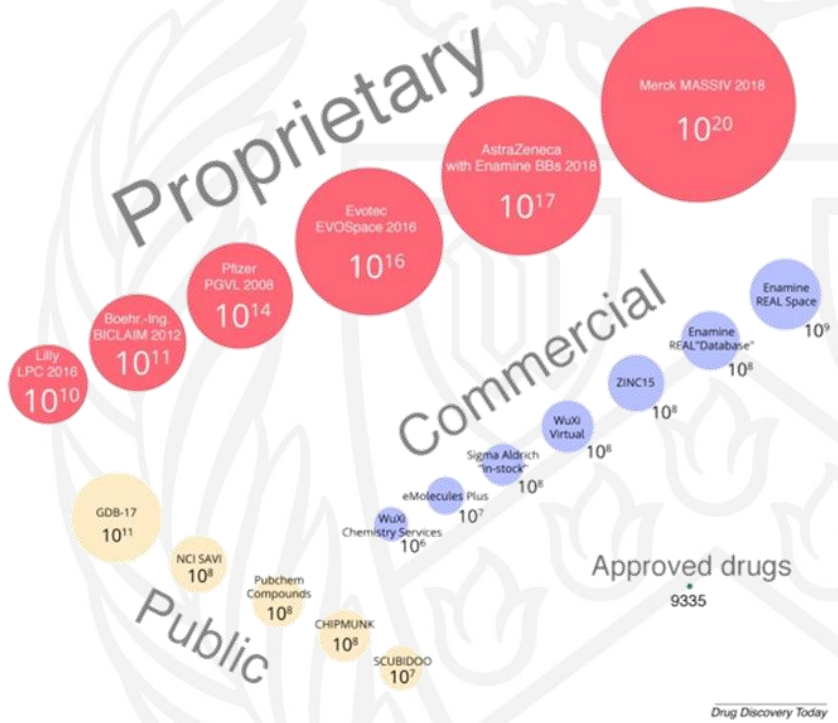




Hachmann, Johannes, et al, Molecular Simulation 44.11 (2018): 921-929

Motivation – Data-driven Discovery Of Molecules

- large datasets of potential structures available for chemical/material discovery
- lab/MD-driven high-throughput screening (HTPS) of datasets impractical
- can we leverage experimental data to speed up screening?



Hoffmann, T, and M. Gastreich, Drug discovery today 24.5 (2019): 1148-1156

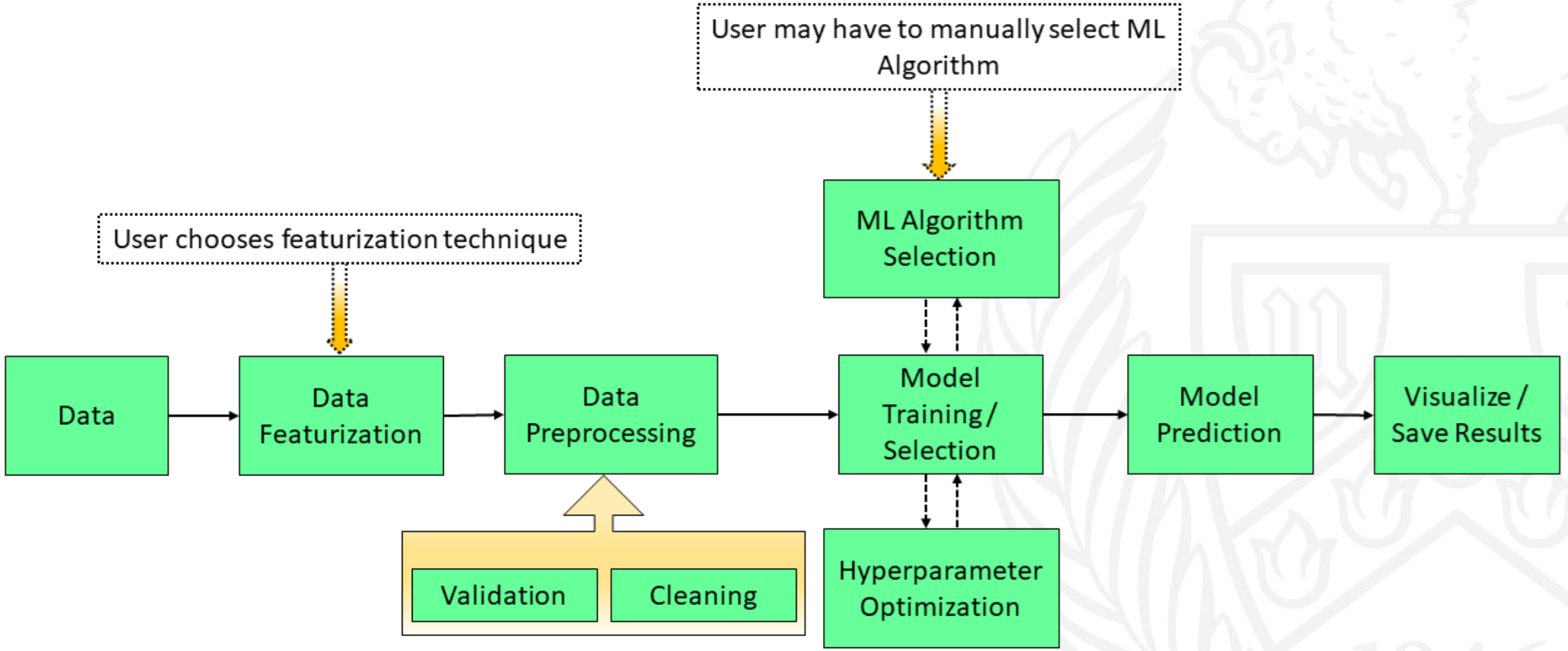
Motivation – Data-driven Discovery Of Molecules

- Machine Learning (ML) - map molecular features to target property
- able to pre-screen molecules at high speed & with good accuracy
- promising candidates can be further verified by MD/lab synthesis

Figure 3. Structure–property relationship mapping *via* machine learning: The structures of the n compounds of a given data-set are represented in an m-dimensional descriptor space, resulting in an $n \times m$-dimensional descriptor matrix D. This descriptor matrix is to be connected to the known properties of the given compounds, which are written as the property vector P of length n. We are seeking the complex mapping function f that encapsulates the structure–property relationship $P = f(D)$ by means of machine learning.

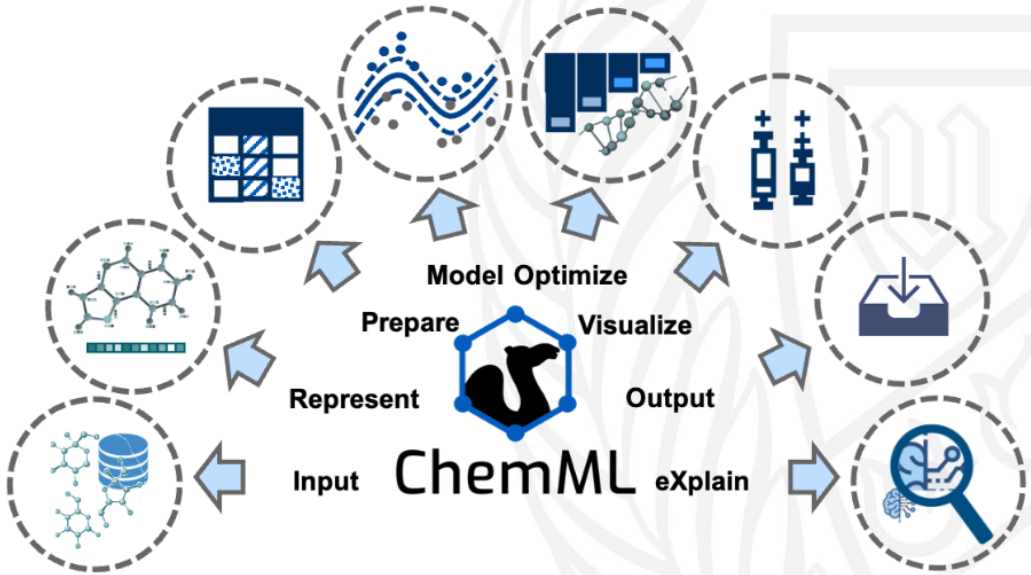
Hachmann, Johannes, et al, Molecular Simulation 44.11 (2018): 921-929

ML-driven Discovery Workflow



ChemML: ML For Chemical And Materials Data

- aim: facilitate access to ML tools for experts & non-experts in chemical & materials domain
- end-to-end pipeline for modeling chemical properties
- leverage your expertise to frame the problem/data, *ChemML* handles the ML behind-the-scenes



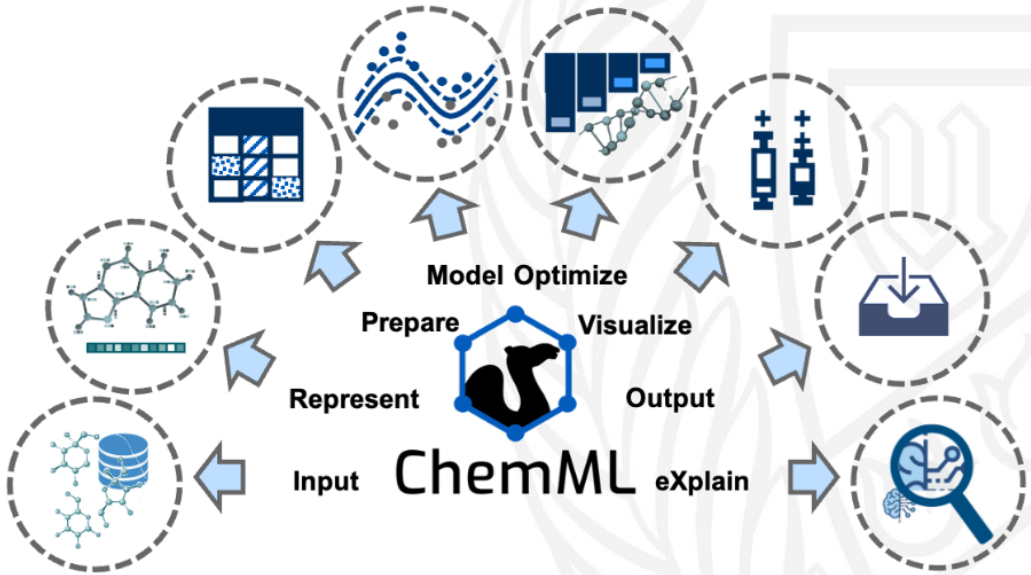
ChemML workflow

ChemML: Demo I

- aim: introduce modules, familiarize with software

ChemML: Challenge with pipeline

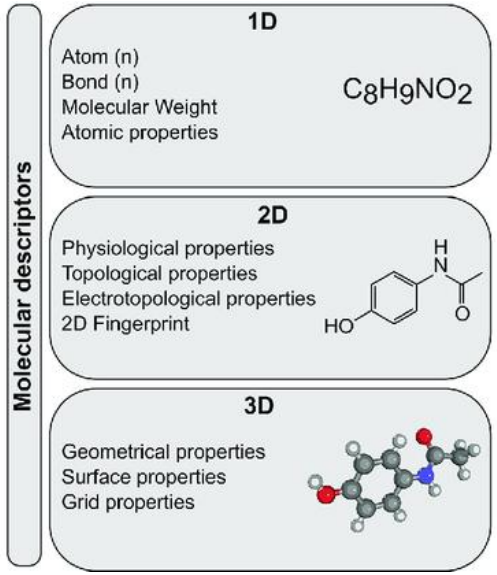
- aim: facilitate access to ML tools for experts & non-experts in chemical & materials domain
- end-to-end pipeline for modeling chemical properties
- can generate robust models *when provided with specific representation technique and model type*

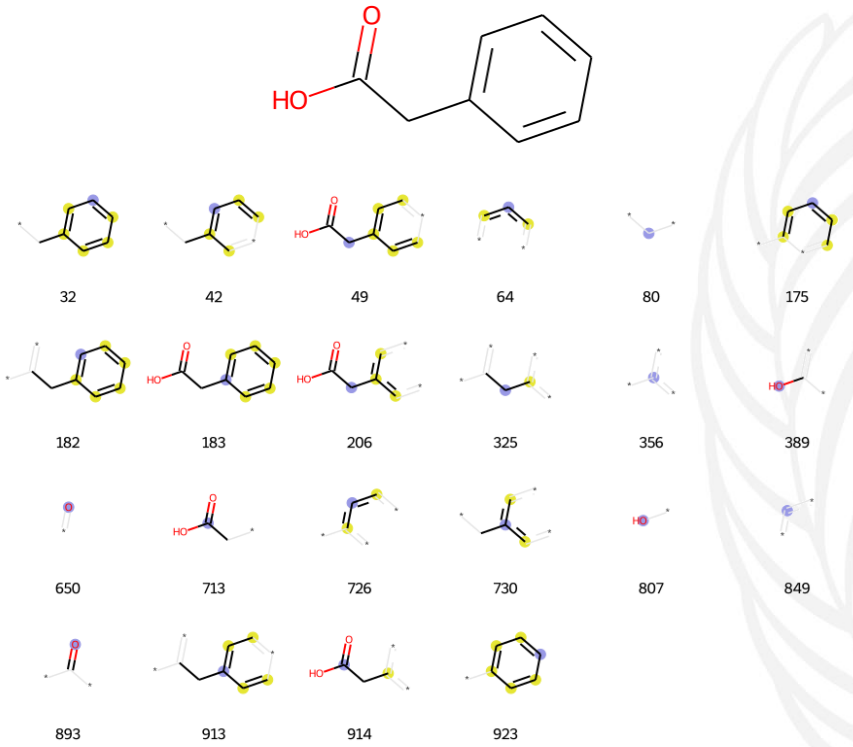


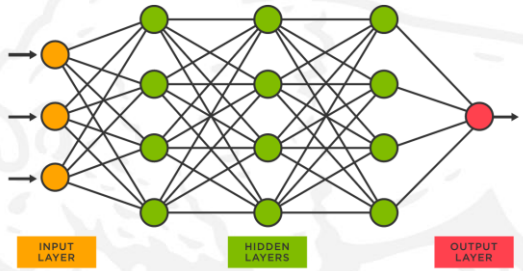
ChemML workflow

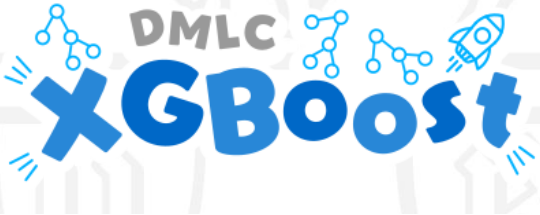
ChemML: Challenge with pipeline

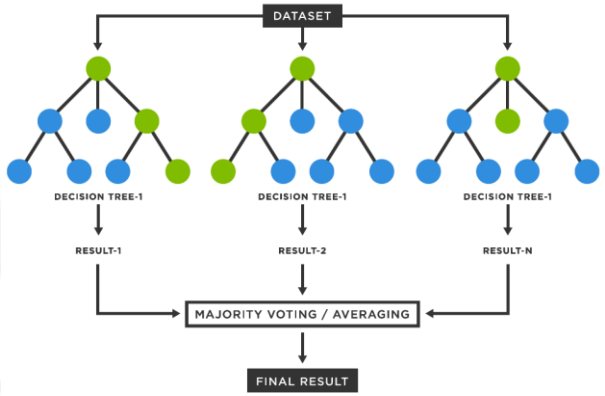
- expertise needed in representation of molecules & machine learning models
- can we make this easier for newer users?





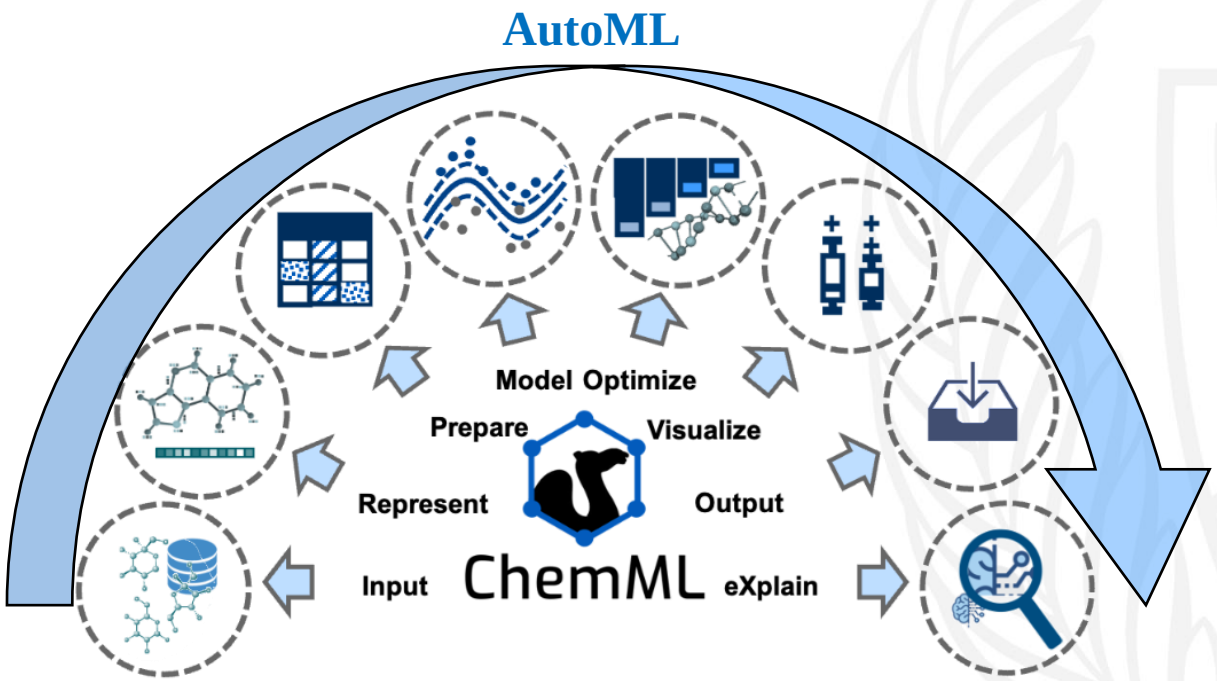






AutoML: An Easier ML Modeling Experience

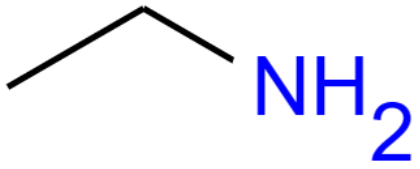
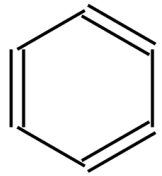
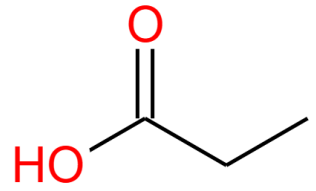
- allows usage of *ChemML* pipeline with minimal user input
- no expertise in ML required, minimal programming necessary
- results are reproducible using open-source ML libraries



ChemML workflow (using AutoML)

Building an ML model using *AutoML*: Input

- input DataFrame:
  - **smiles** strings
  - **target** property(ies)

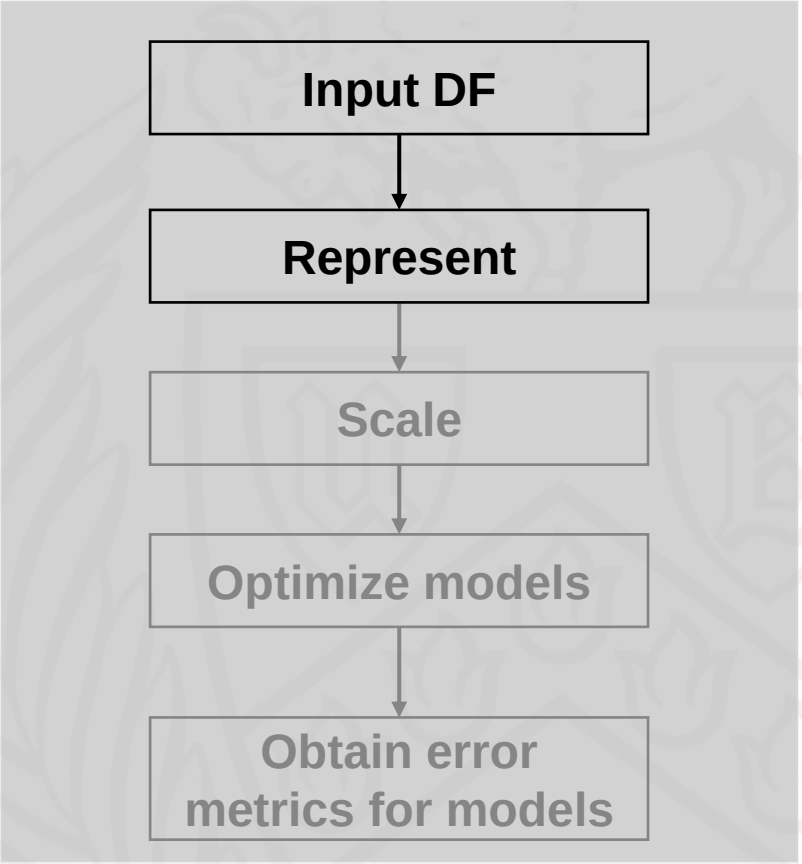
Molecule	SMILES code	Structure
Ethylamine	NCC	
Benzene	c1ccccc1	
Propionic Acid	CCC(=O)O	

Simple Molecular Input Line Entry (SMILES) code examples

smiles	density_Kg/m3
c1scc(n1)c1ncc(s1)c1c(ccc2c1cccc2)c1cncs1	1328.35
S1CC(SC(C1)C1CCCC1)c1cocc1c1ccccc1	1143.80
N1CNC(NC1)c1sccc1C1(CSCCS1)c1cccs1	1351.33
c1nc(c(s1)c1nsnc1)c1csc(n1)c1cccs1	1466.57
Oc1ccc2c(c1c1cocc1)c(ccc2)c1ccccn1	1207.67
Oc1ccc(s1)N1CNC(NC1)c1cnccn1	1355.15
n1csc(n1)C1(CCCC1)c1csc(c1)c1scnc1	1311.30
Oc1ccoc1c1ccnc(c1)c1ccc2c(c1)cccc2	1226.34
SC1CCC(C1c1cnccn1)C1CSCCS1	1238.45
o1ccc(c1)c1sc(cc1c1ccsc1)c1cccc2c1cccc2	1246.33
c1ccc(nc1)c1csc(n1)Oc1ccc2c(c1)cccc2	1250.51
c1cnc(cn1)c1nccnc1c1ccc(o1)c1nccs1	1321.35
C1NC(NC(N1)c1cc(cc2c1cccc2)c1ccccn1)C1CCCC1	1106.21
Sc1oc(c(c1)c1cnccn1)N1CNCNC1	1340.03

Building an ML model using *AutoML*: Supported Featurization

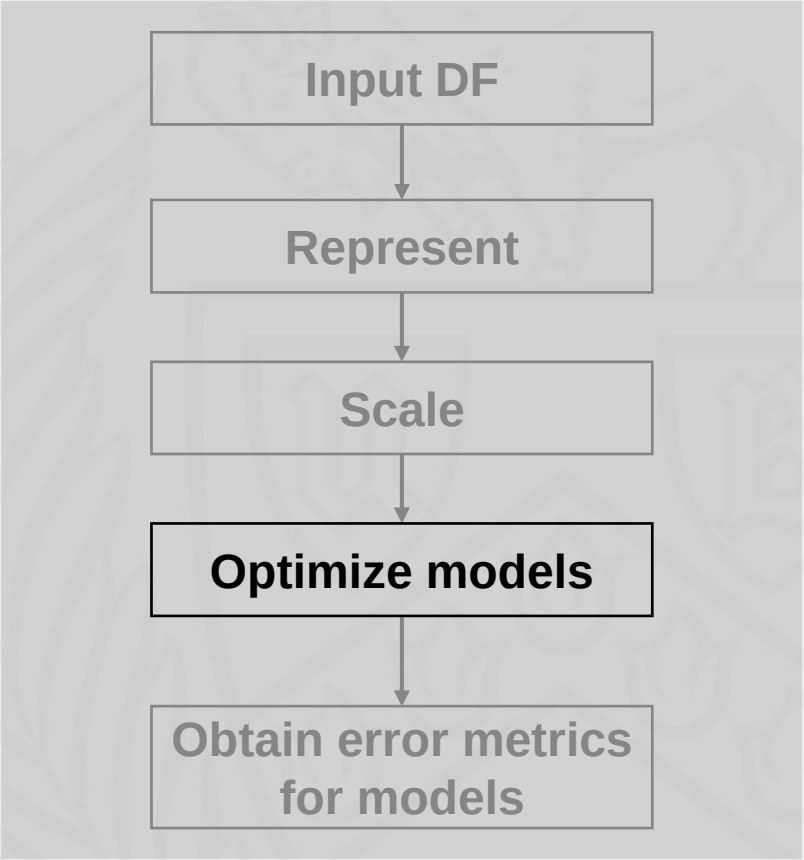
- **screen** through **multiple featurization techniques**:
  - coulomb matrix
  - RDKit fingerprints
    - Morgan fingerprint (ECFP)
    - MACCS keys
    - hashed topological torsion
    - hashed atom pair
  - RDKit/Mordred descriptors
  - **custom features also supported!**



AutoML workflow

Building an ML model using *AutoML*: Supported Models

- **screen** through **multiple ML models**:
  - linear models (Ridge, LASSO, ElasticNet)
  - SVR (under 1k datapoints)
  - decision tree regression
  - neural networks (MLPRegressor/PyTorch/TensorFlow)
  - gradient boosting models (XGBoost/LightGBM)*
- Optimization technique – Genetic Algorithm
  - time efficient, accurate

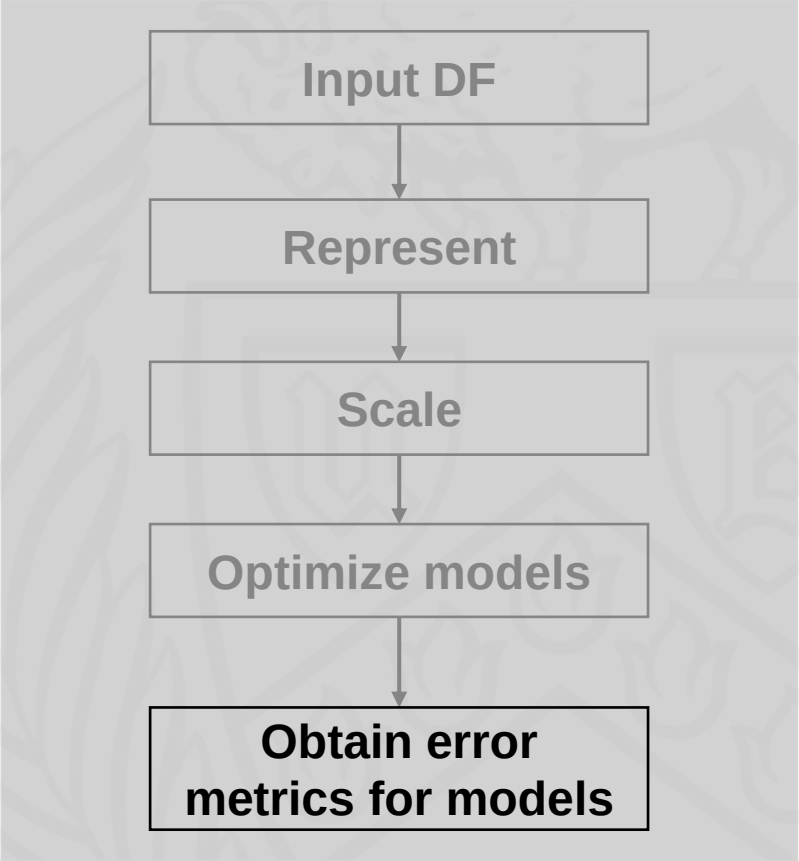


AutoML workflow

Pradhan, A; University at Buffalo, 2024

Building an ML model using *AutoML*: Supported Error Metrics

- **User can evaluate** various **error metrics**
  - ME
  - MAE
  - MSE
  - RMSE
  - MSLE
  - RMSLE
  - MAPE
  - MaxAPE
  - RMSPE
  - MPE
  - MaxAE
  - R^2



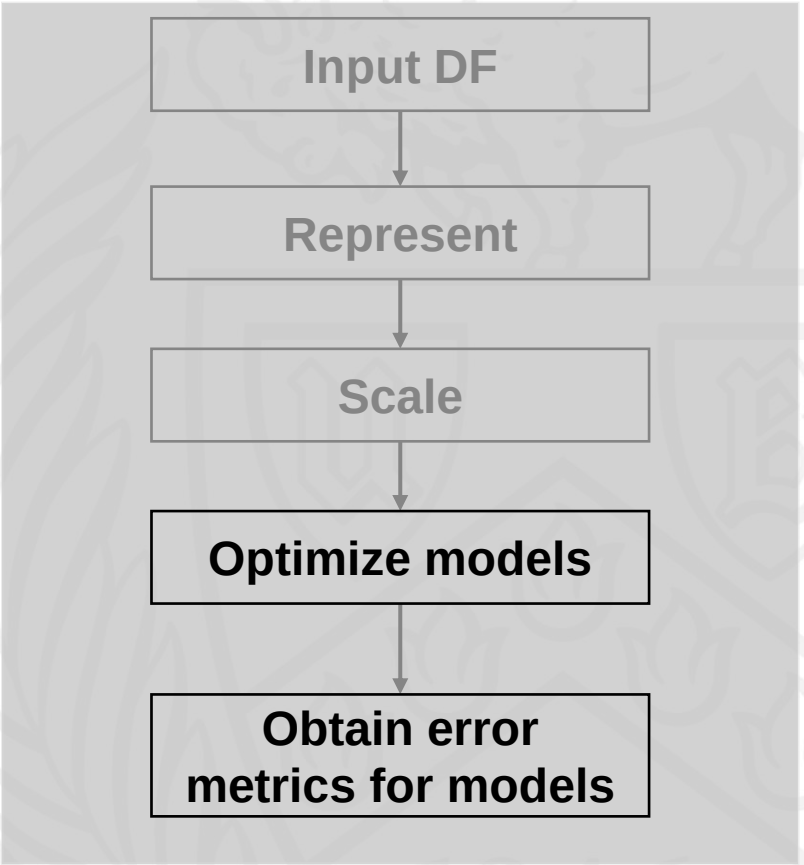
AutoML workflow

Pradhan, A; University at Buffalo, 2024

Building an ML model using *AutoML*: For Classifiers

- **screen** through **multiple classification ML models**:
  - LogisticRegression
  - DecisionTreeClassifier
  - RandomForestClassifier
  - SVC
  - KNeighborsClassifier

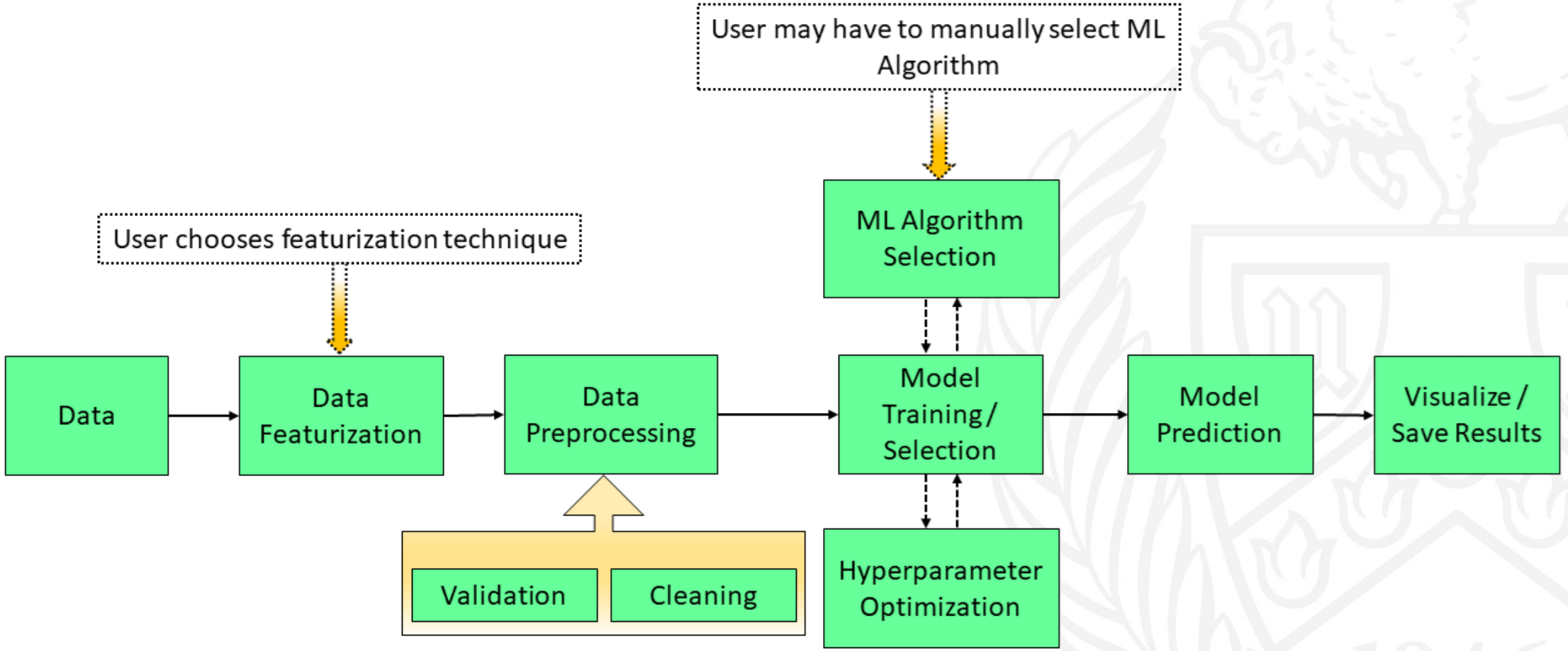
- **user can evaluate** various **error metrics**
  - Accuracy
  - Precision
  - Recall
  - F_1 score



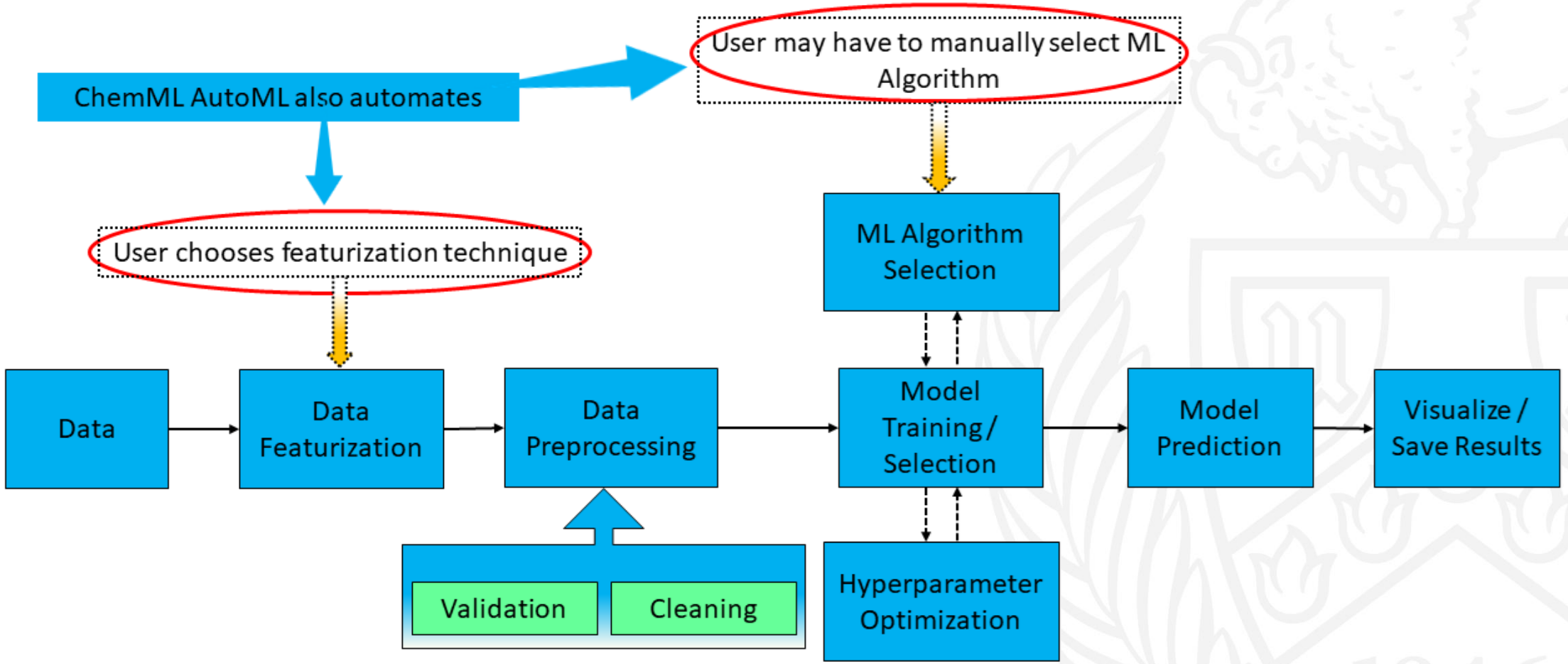
AutoML workflow

Pradhan, A; University at Buffalo, 2024

ML-driven Discovery Workflow



AutoML-Enhanced Discovery Workflow



ChemML AutoML workflow

Results of *AutoML* workflow

Load data

importing libraries and dataset

```
1 import pandas as pd
2 import numpy as np
3 from chemml.chem import Molecule
4 from chemml.datasets import load_organic_density
```

```
1 molecules, target, dragon_subset = load_organic_density()
2 df=pd.concat([molecules, target], axis=1)
3 df.head()
```

]:

	smiles	density_Kg/m3
0	C1CSC(CS1)c1ncc(s1)CC1CCCC1	1184.64
1	Oc1nccnc1c1coc(c1)c1cnccn1	1333.85
2	N1CN(CN(C1)c1cncs1)c1ccc(cn1)c1cocc1	1332.41
3	OC1(CSCCS1)c1ccoc1O	1370.11

starting df
- smiles
- targets

Run autoML

- Easy to import class
- Easy to initialize class
- Easy to call function

```
1 from chemml.autoML import ModelScreener
2 MS = ModelScreener(df, target="density_Kg/m3", featurization=True, smiles="smiles",
3                    screener_type="regressor", output_file="testing.txt")
4 scores = MS.screen_models(n_best=4)
```

```
featurizing molecules in batches of 62 ...
500/500 [==================================================] - 2s 3ms/step
Merging batch features ...    [DONE]

Running model no:  1 ; Name:  RandomForestRegressor
RandomForestRegressor : GeneticAlgorithm - complete
Model: RandomForestRegressor
GA time(hours):  0.14414592577351465


scores_list:  [         ME        MAE         MSE       RMSE      MSLE
0  3.766114  33.290882  1855.76677  43.078612  0.001197  0.034601  2.65

     MaxAPE      RMSPE       MPE      MaxAE   deltaMaxE  r_squared
0  9.274192  3.500977  0.126405  102.68927  184.648282   0.693355  77.7
```

output
- error metrics
- model type
- parameters
- featurization

ME	MAE	MSE	RMSE	MSLE	RMSLE	MAPE	MaxAPE	RMSPE	MPE	MaxAE	deltaMaxE	r_squared	time(seconds)	Model	parameters	Feature
0.00	0.02	0.00	0.02	0.00	0.01	1.14	2.67	1.41	0.03	0.04	0.08	0.74	2.20	Ridge	{'alpha': 112.8, 'copy_X': True, 'fit_intercept': True, 'max_iter': None, 'positive': False, 'random_state': None, 'solver': 'auto', 'tol': 0.0001}	rdkit_descriptors

ChemML: Demo II

- aim: demonstrate *AutoML* workflow