Project Report

Predicting Enthalpy of Vaporization from Molecular Descriptors

Course: CHE657 - Machine Learning For Chemical Engineers

Group Number - 3

Ayush Omer – ayusho23@iitk.ac.in
Harshvardhan Gaur – hgaur23@iitk.ac.in
Prince Yadav – princey23@iitk.ac.in
Khushi Jain – khushija23@iitk.ac.in
Om jee singh – omj23@iitk.ac.in

IIT Kanpur

Instructor: Prof. Salman Khan

Date: 10 November 2025

1. Abstract

Accurate prediction of enthalpy of vaporization (ΔH_{vap}) is essential for chemical process design and understanding intermolecular interactions. This work employs molecular descriptors generated from RedKit to predict ΔH_{vap} for a range of volatile organic compounds (VOCs). After preprocessing and scaling, multiple regression algorithms including Linear Regression, Random Forest, Support Vector Regression, XGBoost, LightGBM, CatBoost, Gradient Boosting, Ridge, and Lasso were evaluated. Among them, CatBoost achieved the best predictive accuracy ($R^2 = \ldots$, RMSE = $\ldots$). The results demonstrate the effectiveness of ensemble learning techniques in capturing complex nonlinear relationships between molecular structure and vaporization enthalpy.

2. Introduction

The enthalpy of vaporization (ΔH_{vap}) quantifies the energy required to transform a liquid into its vapor at constant pressure and is a key parameter in thermodynamics, chemical engineering, and environmental modeling. Reliable estimation of ΔH_{vap} supports process design, safety assessments, and understanding of molecular interactions, especially for volatile organic compounds (VOCs).

Experimental determination of ΔH_{vap} is accurate but resource-intensive, motivating the use of data-driven predictive approaches. With advances in cheminformatics, molecular descriptors—quantitative representations of molecular structure—allow machine learning (ML) models to learn structure–property relationships directly from data. This eliminates the need for complex theoretical derivations or extensive laboratory measurements.

In this project, molecular descriptors generated via RedKit were used to develop predictive models for ΔH_{vap}. A range of ML algorithms—from simple linear models to advanced ensemble methods such as Random Forest, Gradient Boosting, XGBoost, LightGBM, and CatBoost—were compared to evaluate their ability to capture the nonlinear dependence of ΔH_{vap} on molecular structure.

The primary goal of this work is to assess which modeling approach offers the best balance between accuracy, interpretability, and computational efficiency for predicting enthalpy of vaporization from molecular descriptors.

3. Dataset and Feature Generation

3.1 Dataset Source

The dataset was obtained from an online source, containing molecular and thermodynamic information for a set of volatile organic compounds (VOCs). Each entry includes identifiers such as CAS number, SMILES representation, chemical family, and the target property—the enthalpy of vaporization (ΔH_{vap}), denoted as `dvap`.

3.2 Descriptor Generation

Molecular descriptors were generated using **RdKit**, a cheminformatics toolkit that computes a wide range of molecular features. The descriptors encompass 2D, 3D, topological, electronic, and geometric properties, capturing various aspects of molecular structure and behavior. These descriptors provide a comprehensive numerical representation suitable for machine learning models.

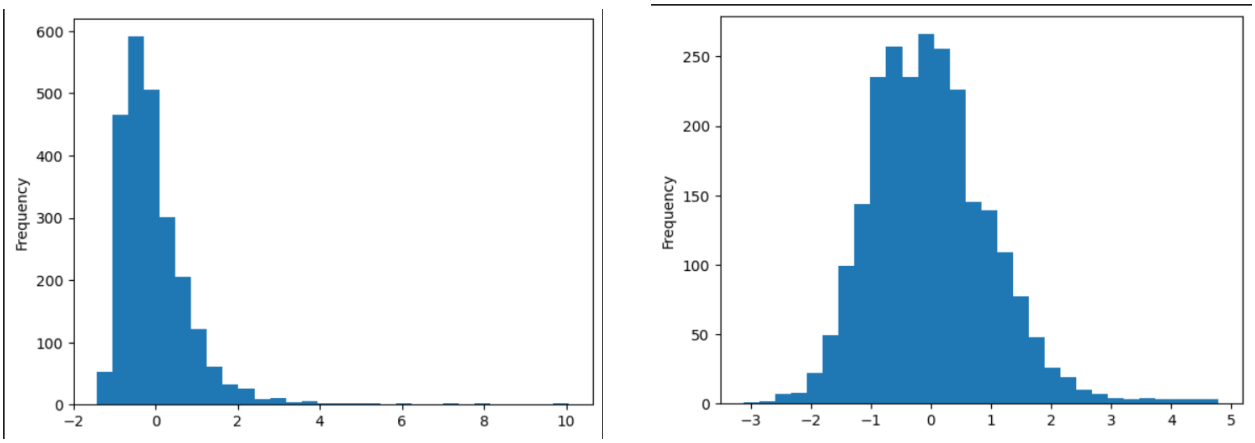
3.3 Data Dimensions

RdKit produced a total of 216 descriptor columns for each compound, along with additional identifiers. These descriptors quantify the structural, electronic, and spatial characteristics of molecules, forming the feature space used for model training.

3.4 Preprocessing Steps

An initial analysis of the target variable (`dvap`) showed that the raw values were heavily right-skewed, with a skewness of approximately 1.8, indicating a long-tailed distribution. To address this, a logarithmic transformation was applied, which substantially reduced the skewness to around 0.4, bringing the distribution much closer to symmetry. Because the log-transformed target exhibited a more normalized and statistically stable distribution, the log-transformed `dvap` values were used for all subsequent modeling.

All feature columns were **standard-scaled** (zero mean, unit variance) to ensure uniformity across descriptors. The dataset was then divided into **training and test sets (70:30)** to evaluate model generalization.



(a) Original distribution of `dvap` (b) Log-transformed distribution of `dvap`

Figure 1: Comparison of target variable distributions before and after log transformation.

The dataset initially contained **216 molecular descriptors**, many of which were highly correlated and redundant. To identify the most informative molecular descriptors, Principal Component Analysis (PCA) was applied to the standardized feature matrix. PCA decomposes the data into orthogonal components that capture maximum variance, and the contribution (loading) of each original feature to these components indicates its overall influence on the dataset structure. After fitting PCA, the absolute loadings were weighted by the explained variance ratio of each component, and these weighted contributions were summed across all components to compute a global importance score for every feature. The features were then ranked based on this importance measure, and the top 20–30 descriptors with the highest cumulative contribution to the principal components were selected. This ensures that the retained features are those most responsible for variance in the data while reducing redundancy and dimensionality.

4. Methodology

This study applies multiple machine learning algorithms to model the relationship between molecular descriptors and the enthalpy of vaporization (ΔH_{vap}). Each model type captures different forms of structure–property relationships, from simple linear dependencies to complex nonlinear interactions.

4.1 Linear Regression

Linear Regression serves as the baseline model, assuming a linear relationship between features and the target variable:

$$\hat{y} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

where x_i are molecular descriptors and β_i are the learned coefficients. The model minimizes the sum of squared errors (SSE) between predicted and actual values. It provides interpretability but often underfits nonlinear data.

4.2 Ridge and Lasso Regression

Both are regularized linear models that improve generalization by penalizing large coefficients.

Ridge Regression (L2 penalty):

$$\text{Loss} = \sum (y_i - \hat{y}_i)^2 + \lambda \sum \beta_i^2$$

Shrinks coefficients toward zero but does not eliminate them.

Lasso Regression (L1 penalty):

$$\text{Loss} = \sum (y_i - \hat{y}_i)^2 + \lambda \sum |\beta_i|$$

Performs feature selection by driving some coefficients exactly to zero.

These models help control overfitting and handle multicollinearity among descriptors.

4.3 Support Vector Regression (SVR)

SVR is based on the Support Vector Machine (SVM) framework. It finds a regression line (or hyperplane) that fits the data within a tolerance margin ϵ while minimizing the impact of outliers:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad |y_i - (w \cdot x_i + b)| \leq \epsilon$$

Using kernel functions (e.g., radial basis function, RBF), SVR captures complex, nonlinear relationships effectively.

4.4 Random Forest Regressor

Random Forest is an ensemble of decision trees, where each tree is trained on a bootstrap sample of the data and uses a random subset of descriptors at each split. The final prediction is the average of all tree outputs:

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

This method reduces variance and improves robustness while handling nonlinear dependencies and feature interactions.

4.5 Gradient Boosting Regressor

Gradient Boosting builds trees sequentially, where each new tree attempts to correct the residuals (errors) of the previous ensemble. The model minimizes a differentiable loss function (e.g., mean squared error) by gradient descent:

$$F_m(x) = F_{m-1}(x) + \eta h_m(x)$$

where $h_m(x)$ is the new weak learner and η is the learning rate controlling the contribution of each tree.

4.6 XGBoost, LightGBM, and CatBoost

These are advanced implementations of gradient boosting that improve efficiency, speed, and generalization:

- **XGBoost** employs regularization, tree pruning, and parallel computation to enhance performance and prevent overfitting.
- **LightGBM** uses histogram-based splitting and leaf-wise tree growth for faster training and better handling of large feature spaces.
- **CatBoost** is optimized for categorical data and mitigates overfitting through ordered boosting and efficient handling of feature combinations.

These models are powerful for capturing nonlinear, high-dimensional relationships present in molecular descriptor data.

4.7 How Trees Work in Boosting Algorithms

Boosting methods build an ensemble of *regression trees*, where each tree corrects the errors of the previous trees. The tree-building process in boosting follows these steps:

1. Initialize a Base Prediction: Boosting begins with a simple constant prediction for all samples:

$$F_0(x) = \arg\min_c \sum_i (y_i - c)^2$$

For mean squared error loss, this value is simply the mean of the target variable.

2. Compute Residuals (Pseudo-Targets): Each subsequent tree is trained on the residuals, which represent the errors of the model so far:

$$r_i^{(m)} = y_i - F_{m-1}(x_i)$$

These residuals act as new target values.

For general loss functions, boosting uses the first derivative (gradient) and sometimes the second derivative (Hessian) instead of simple residuals.

3. Build a Regression Tree on the Residuals: A decision tree is trained to predict these residuals. At each split:

- All features and possible split points are evaluated.
- The split that reduces the loss the most (e.g., sum of squared residuals) is chosen.
- This recursion continues until a stopping criterion is reached (maximum depth, minimum samples, or minimum gain).

Each leaf of the tree is assigned an optimal value:

$$w_j = \frac{\sum_{i \in leaf_j} r_i^{(m)}}{\sum_{i \in leaf_j} h_i}$$

where h_i represents the Hessian for second-order methods (as used in XGBoost and LightGBM).

4. Update the Model: The new tree contributes a small correction to the model:

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x)$$

where η is the learning rate and $h_m(x)$ is the prediction of the m-th tree.

5. Repeat for Many Trees Over many iterations, the ensemble becomes increasingly accurate because each tree focuses on the remaining errors of the previous trees.

4.7 Model Evaluation

All models were trained on 70% of the dataset and evaluated on the remaining 30%. Performance was compared using:

- **Coefficient of Determination (R^2)**: Measures explained variance.
- **Root Mean Square Error (RMSE)**: Penalizes larger errors.
- **Mean Absolute Error (MAE)**: Captures average deviation.

5. Results and Discussion

The performance of different machine learning models for predicting ΔH_{vap} is summarized in Tables below. Models were evaluated using R^2, RMSE, and MAE on both training and test sets. It is evident that ensemble-based methods outperform linear models and SVR for predicting ΔH_{vap}. Linear Regression, Ridge, Lasso, and ElasticNet provide reasonable accuracy but tend to underfit complex nonlinear relationships. SVR shows the poorest performance, likely due to difficulty capturing high-dimensional descriptor interactions.

Random Forest, Gradient Boosting, XGBoost, LightGBM, and CatBoost all achieve high R^2 values on both training and test sets, with **LightGBM and CatBoost achieving the lowest test RMSE and MAE**, indicating strong predictive accuracy and generalization. While XGBoost has the highest training accuracy, its test RMSE is slightly higher than LightGBM and CatBoost, suggesting minor overfitting.

Overall, **CatBoost and LightGBM are the most suitable algorithms** for robust prediction of ΔH_{vap} from molecular descriptors, effectively capturing complex nonlinear dependencies.

Model	Train_R2	Train_RMSE	Train_MAE	Test_R2	Test_RMSE	Test_MAE
Linear Regression	0.9518	5.845	3.845	0.9408	6.786	4.637
RandomForestRegressor	0.9942	2.025	1.231	0.9586	5.674	3.410
GradientBoostingRegressor	0.9837	3.397	2.515	0.9609	5.520	3.650
SVR	0.6225	16.350	9.394	0.6725	15.965	9.389
MLP Regressor	0.9432	6.344	3.616	0.9326	7.242	4.171
Ridge	0.9493	5.989	3.860	0.9430	6.659	4.524
Lasso	0.9484	6.046	3.992	0.9418	6.732	4.620
ElasticNet	0.9473	6.111	3.970	0.9428	6.670	4.557
XGBoost	0.9994	0.627	0.395	0.9629	5.372	3.075
LightGBM	0.9993	0.694	0.373	0.9696	4.866	2.829
CatBoost	0.9976	1.309	0.973	0.9707	4.773	2.657

Table 1: Model Performance Comparison for log enthalpy

Model	Train_R2	Train_RMSE	Train_MAE	Test_R2	Test_RMSE	Test_MAE
LinearRegression	0.975854	5.113132	3.594191	0.964290	5.846174	3.964398
RandomForestRegressor	0.996004	2.080032	1.260634	0.963405	5.918154	3.525481
GradientBoostingRegressor	0.988377	3.547474	2.602360	0.967661	5.563423	3.740417
SVR	0.458257	24.219476	10.445443	0.577559	20.107520	9.948005
MLPRegressor	0.976308	5.064857	3.455595	0.959738	6.207591	3.938500
Ridge	0.975106	5.191793	3.616063	0.962874	5.960970	4.017953
Lasso	0.972752	5.431716	3.797424	0.958608	6.294094	4.307391
ElasticNet	0.972675	5.439412	3.785200	0.959013	6.263251	4.282186
XGBRegressor	0.999613	0.647200	0.412027	0.963851	5.881953	3.197689
LGBMRegressor	0.996907	1.830153	0.593529	0.965654	5.733452	3.196987
CatBoostRegressor	0.998465	1.289224	0.954972	0.974057	4.982976	2.836754

Table 2: Model Performance Comparison for enthalpy

6. Conclusion and Future Work

This study aimed to predict the enthalpy of vaporization (H_{vap}) of organic compounds using molecular descriptors generated by RedKit. Various regression models—including Linear Regression, Ridge, Lasso, Random Forest, SVR, Gradient Boosting, XGBoost, LightGBM, and CatBoost—were trained on standardized data. Among these, CatBoost and LightGBM achieved the best performance, indicating their strength in capturing nonlinear relationships between molecular structure and H_{vap}.

In future work, the model can be improved by expanding the dataset, applying advanced feature selection techniques, and exploring deep learning approaches such as Graph Neural Networks (GNNs) for direct molecular graph learning. Overall, this project demonstrates that machine learning offers an efficient and accurate alternative to experimental or simulation-based estimation of thermodynamic properties like H

6. Rerferences

References

[1] Greg Landrum, *RDKit: Open-source cheminformatics*, http://www.rdkit.org, 20XX.

[2] S. K. Kearsley, A. R. Leach, "Quantitative structure–property relationships," Journal of Chemical Information and Computer Sciences, 199X.

[3] Lundberg, S.M., Lee, S.-I., "A unified approach to interpreting model predictions," Advances in Neural Information Processing Systems, 2017.