Property Predictor

Estimating a molecule's physical properties through the means of Machine Learning.

(No physical Chemistry calculations involved.)

Submitted By:

(102117049) Gudur Krishna Chaitanya

(102117108) Tamogh Nekkanti

(102117009) Rahul Divi

BE Third Year, CSE

Submitted To:

Dr. Arun Singh Pundir

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

TIET, Patiala

Abstract:

To develop a machine learning model capable of estimating the physical properties of molecules solely from their chemical structures. This model aims to predict properties such as boiling point, melting point, solubility, and vapor pressure, among others, without relying on traditional physical chemistry calculations. The goal is to create a predictive tool that can find the physical properties for new molecules based on the data used to train, whose usage is in drug synthesis and discovery, material science and various other fields where understanding molecular behavior is essential.

Problem Description:

In drug discovery and materials science, knowing the properties of molecules is crucial for finding promising candidates for further study. Traditionally, scientists used experiments and calculations to figure out these properties, which took a lot of time and money. To speed things up, scientists have started predicting molecule behavior using models based on their chemical structures alone. This saves time and resources by focusing on the most promising candidates.

Machine learning (ML) is a powerful tool for this. ML algorithms can look at big datasets of molecules with known properties, learn patterns and relationships between their structures and properties, and then predict the properties of new molecules accurately.

Certain ML algorithms are especially good for this job:

- K-Means: Puts similar molecules together based on their structures, helping spot patterns in the dataset.
- Agglomerative Hierarchical Clustering (Agglo): Finds hierarchical relationships in the data, useful for understanding complex molecular structures.
- DBSCAN: Identifies clusters of different shapes and sizes in a dataset, good at handling outliers and noise.
- t-SNE (t-distributed Stochastic Neighbor Embedding): Turns high-dimensional data into lower-dimensional space, helpful for seeing how molecules relate to each other.

These algorithms are pretty straightforward to use and understand, so even researchers without a ton of ML experience can use them. They can handle big and complicated datasets well, which is important when studying molecular structures.

Besides clustering algorithms, there are also similarity measures like Yule, cosine, and Tanimoto coefficients. These measure how similar molecules are based on their chemical fingerprints or structures, giving insight into how molecules relate to each other.

By using these ML algorithms and similarity measures, researchers can create accurate models for predicting molecule properties. This speeds up drug discovery and materials science research, making it cheaper and more efficient.

Literature Survey:

In recent years, the application of machine learning (ML) techniques in chemistry, particularly in predicting molecular properties from chemical structures, has garnered significant attention from researchers worldwide. This surge in interest stems from the growing demand for efficient and cost-effective methods to estimate physical properties of molecules, such as boiling point, melting point, solubility, and vapor pressure, without relying on traditional physical chemistry calculations.

Several studies have explored the efficacy of ML algorithms in predicting molecular properties based solely on chemical structure information. One notable approach involves the utilization of clustering algorithms, such as K-Means, Agglomerative Hierarchical Clustering (Agglo), and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). These algorithms have demonstrated promising results in grouping molecules with similar structural features, thereby enabling researchers to identify patterns and relationships within large datasets of chemical compounds.

For instance, K-Means clustering has been widely employed to partition molecules into clusters based on structural similarities, facilitating the identification of distinct molecular groups with similar properties. Similarly, Agglomerative Hierarchical Clustering has been effective in uncovering hierarchical relationships among molecules, providing insights into the structural complexity of molecular systems. Additionally, DBSCAN has proven valuable in identifying clusters of varying shapes and sizes, robustly handling outliers and noise present in chemical datasets.

Furthermore, dimensionality reduction techniques, such as t-distributed Stochastic Neighbor Embedding (t-SNE), have been instrumental in visualizing high-dimensional molecular data in lower-dimensional spaces. By transforming complex molecular structures into more interpretable representations, t-SNE enables researchers to gain valuable insights into the relationships and similarities between molecules, facilitating the prediction of their properties.

In addition to clustering algorithms and dimensionality reduction techniques, researchers have explored various similarity measures, including Yule, cosine, and Tanimoto coefficients. These measures quantify the similarity between molecules based on their chemical fingerprints or structural features, providing a deeper understanding of molecular relationships and enhancing the accuracy of property prediction models.

1. Clustering of Chemical Compounds with Similar Molecular Structures for Property Prediction: https://pubs.acs.org/doi/10.1021/acs.jcim.0c01020
2. Machine Learning Models for Predicting Physical Properties of Molecules from Structural Features: https://www.sciencedirect.com/science/article/pii/S2405844020322601
3. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise: https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf
4. Visualizing Data using t-SNE: https://jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf
5. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise(K-Means): https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf
6. Hierarchical Clustering Schemes: https://www.sciencedirect.com/science/article/pii/B9780080510569500148
7. Predicting Physical Properties of Molecules: Comparing K-Means Clustering and Random Forests: https://pubs.acs.org/doi/10.1021/acs.jcim.0c00110
8. Machine Learning Models for Predicting Physical Properties of Organic Compounds: https://pubs.acs.org/doi/10.1021/acs.jcim.0c00756
9. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN: https://dl.acm.org/doi/10.1145/3357384.3357894
10. t-SNE-CUDA: GPU-Accelerated t-Distributed Stochastic Neighbor Embedding: https://arxiv.org/abs/1807.11808

Challenges faced and handling techniques employed:

Data Preprocessing:

1. **Handling Missing Values:** The dataset contains missing values for certain properties of chemical compounds, hindering accurate analysis and model. So the objective is to Develop techniques to address missing data points effectively, ensuring completeness and reliability of the dataset. One approach is to Implement imputation methods or data removal strategies to handle missing values based on the nature and extent of missingness.
2. **Dealing with Outliers:** Outliers in the dataset can distort statistical analyses and machine learning models, leading to biased results.So the objective is to identify and mitigate the impact of outliers on the dataset to ensure robust and accurate analysis. An approach is to apply outlier detection techniques such as Z-score, IQR (Interquartile Range), or robust statistical methods to identify and address outliers appropriately.

3. **Standardizing Chemical Names:** The dataset contains chemical compounds with diverse naming conventions, leading to inconsistencies in representation. The objective is to standardize the naming conventions for chemical compounds to facilitate uniformity and compatibility across the dataset. An approach is to use mapping techniques to consolidate different naming variations into a standardized format for each compound using string manipulations.
4. **Converting Non-Numerical Descriptions and Unit Normalization:** Physical properties of chemical compounds may be described in non-numeric terms, complicating quantitative analysis. The objective Convert non-numeric descriptions of physical properties into numerical quantities for consistent analysis and modeling. The approach that we used is we separated the numerical values from the string representations and using the units mentioned for each quantity we normalized units of all the data points to a single unit.
5. **Addressing Inconsistencies in Representations:** Chemical compounds are represented in various formats (e.g., SMILES, MOL files), leading to inconsistencies and compatibility issues. The objective is to standardize the representations of chemical compounds to ensure uniformity and compatibility across the dataset. We used the RDKit library of python to validate and normalize all the molecular representations to SMILES, eventually to motif and RDKit object representations for algorithmic processing.

Model selection, training and testing:

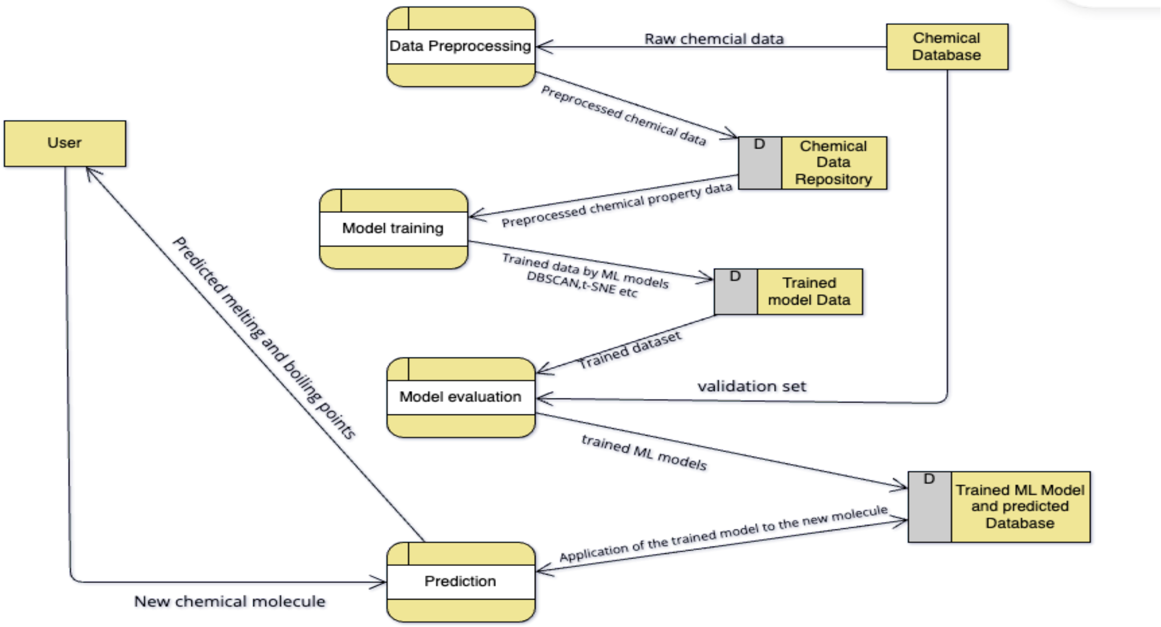
1. **Clustering-Based Algorithms:** Selecting the appropriate clustering algorithm for model selection and training poses a challenge due to the diversity and complexity of the chemical compounds in the dataset. We explored various clustering algorithms, including Agglomerative Hierarchical Clustering, K-Means, DBSCAN, and t-SNE, to identify clusters of similar molecules based on their properties.
2. **Optimization of K in K-Means Clustering:** Determining the optimal number of clusters (k) in K-Means clustering is crucial but challenging. We utilized the elbow method to optimize k, ranging from 5 to 20 clusters, by evaluating the inertia for different values of k.
3. **Similarity Metrics Selection:** Selecting the most suitable similarity metric for measuring the similarity between molecules is essential for accurate prediction. We considered various similarity metrics such as Yule, cosine, and Tanimoto coefficients to identify the optimal metric that best captures the relationships between molecules.
4. **String Similarity vs. Molif Similarity:** Choosing between string similarity and molif similarity methods for comparing molecules presents a dilemma. We evaluated the advantages and disadvantages of each approach and selected the most appropriate method based on its effectiveness for our dataset.

5. **Centroid Calculation Difficulty:** Calculating the centroid molecule in clustering algorithms like K-Means is not feasible due to the complexity of the data. Instead of calculating centroids, we averaged out the similarity between all molecules in a cluster to the given test molecule, providing an estimate of the properties of the test molecule based on its similarity to other molecules in the cluster.
6. **DBSCAN Outlier Handling:** DBSCAN clustering considers a majority of the data points as outliers, resulting in a single outlier cluster (-1). We recognized the limitations of DBSCAN for our dataset.
7. **T-sne and manual interpretation of clusters:** Interpreting clusters generated by clustering algorithms manually can be time-consuming and subjective, requiring human intervention for analysis. Additionally, the visualization of high-dimensional data poses challenges for understanding the underlying patterns effectively. Therefore, there is a need to develop a method to interpret clusters in a non-manual manner and utilize visualization techniques to automate the further clustering process.

Novelty

1. The problem statement addresses the inefficiencies in traditional methods of estimating molecule properties, offering a data-driven approach to prediction, thus bridging the gap between chemical structure and physical properties.
2. The approach used, enables the identification of complex patterns and relationships within large datasets, facilitating accurate property estimation using simple clustering ML algorithms.
3. Usage of physical chemistry calculations, quantum chemistry calculations, physical experimentation requires a lot of compute power, human intervention, manual processing, cost and theoretical understanding, upon which too the results can't assure understanding of structure vs. properties as ML does.
4. By the incorporation of diverse similarity metrics and optimization techniques, the approach achieves higher prediction accuracy, surpassing conventional methods.
5. The problem statement assists chemists, researchers, and other computational chemistry enthusiasts in comprehending the intricate patterns underlying molecular structure-property mapping. It facilitates a clear understanding of drugs and their properties before testing, potentially yielding insightful information. This, in turn, aids in comprehending the behavior of compounds, prompting cautious consideration.

Data flow Diagram:


Data Preprocessing
Raw chemcial data
Chemical Database
User
Preprocessed chemical data
D
Chemical Data Repository
Model training
Preprocessed chemical property data
Predicted melting and boiling points
Trained data by ML models DBSCAN,t-SNE etc
D
Trained model Data
Trained dataset
Model evaluation
validation set
trained ML models
D
Trained ML Model and predicted Database
Application of the trained model to the new molecule
Prediction
New chemical molecule

The above diagram showcases how the training data is passed through a model ,how the predictions are made and finally how we evaluate our test data.

Dataset Description:

Elaborating on Dataset Creation for Chemical Property Prediction Here's a more detailed breakdown of the process you described for creating your chemical property prediction dataset:

Part 1: BradleyDoublePlusGoodMeltingPointDataset.csv

We utilized a pre-built dataset named "BradleyDoublePlusGoodMeltingPointDataset.csv," containing molecules with their corresponding melting points.

Preprocessing: SMILES Strings: SMILES strings were extracted from the dataset, providing a compact representation of each molecule's structure. Molif: The molecules were converted to a Machine-Readable Format (MRF) like MOL files using RDKit, facilitating programmatic manipulation of molecular structures.

Data Type Validation: All data types, including strings and integers, were validated to ensure correct formatting for further processing.

Storage: The preprocessed data was stored in the Pickle format, an efficient Python-specific format for complex data structures.

Part 2: Sigma-Aldrich Website Scraping

Data was scraped from the Sigma-Aldrich chemical supplier website.

Target Chemicals: Emphasis was placed on common acids, bases, and salts.

Preprocessing: Name Cleaning: The names of scraped acids, bases, and salts were cleaned and standardized, involving the removal of inconsistencies, typos, and formatting variations. Property Retrieval with RDKit: RDKit was used to retrieve relevant chemical properties for each molecule, such as melting point, boiling point, and solubility.

External Data Source (NCBI) for Physical property data: Physical property data, including boiling point, melting point, and solubility, was scraped from the National Center for Biotechnology Information (NCBI) website. Preprocessing of Properties: All property values were standardized to numerical format (e.g., converting strings to floats) and ensured consistent units, essential for machine learning models requiring standardized numerical data.

Combining the Datasets: The two preprocessed datasets (BradleyDoublePlusGoodMelting -PointDataset and scraped Sigma-Aldrich data) were merged. This resulted in a final dataset containing SMILES strings/MOL files, retrieved chemical properties (from RDKit), and scraped physical properties (from NCBI) with consistent units. This combined dataset could then be used to train a machine learning model for predicting chemical properties of new molecules based on their SMILES string representation.

Algorithms Used:

The main algorithms we have employed to solve the problem statement are all unsupervised methods.The algorithms are-Kmeans,Agglomerative clustering,Dbscan and t-sne.Let us see what these methods are, how the results produced between them varied and which is the best method. But before that I would like to draw attention to a few distance matrices used in our project.:

1.Yule Distance: Yule distance is a measure of dissimilarity between two probability distributions based on their overlap. The Yule distance between two probability distributions, P and Q is given by: The distance ranges from 0 to 1, with 0 indicating that the two distributions are identical and 1 indicating that they have no overlap.

2.Cosine Distance: Cosine Distance is but 1-Cosine similarity where we can calculate cosine similarity as:
Cosine Similarity(Cs)= $x.y/\sqrt{x.x}\sqrt{y.y}$ where ‘.’ is the dot product between the two vectors

3.Tanimoto Distance: The Tanimoto distance between any two ID 's is defined as 1 minus the number of unique elements in the intersection of their FP 's, divided by the number of unique elements in the union of their FP 's

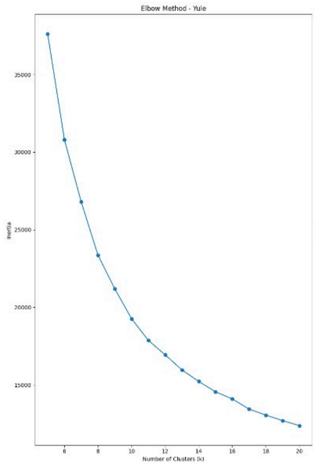
4.Threed Distance: The formula for calculating the distance between two points in three-dimensional space is: distance = sqrt ((x2 - x1) 2 + (y2 - y1) 2 + (z2 - z1) 2) Where: x1, y1, z1: the coordinates of the first point x2, y2, z2: the coordinates of the second point sqrt: the square root function.

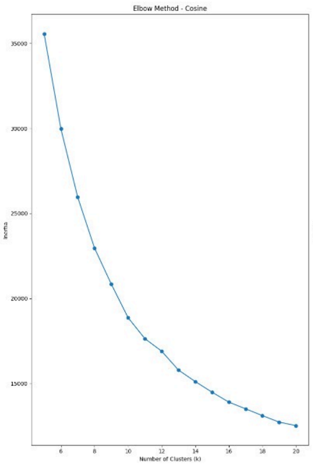
Now that we have defined what are the different distance matrices we have used let us employ each of them in our clustering algorithms:

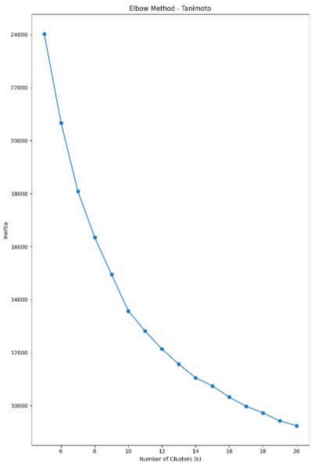
1.K-Means:

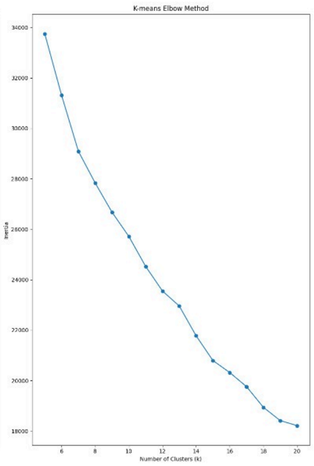
K means clustering, assigns data points to one of the K clusters depending on their distance from the center of the clusters. It starts by randomly assigning the clusters centroid in the space. Then each data point is assigned to one of the clusters based on its distance from the centroid of the cluster. After assigning each point to one of the clusters, new cluster centroids are assigned. This process runs iteratively until it finds a good cluster.

Firstly we shall take a distance matrix and fit a k-means algorithmic model on the training data so as to get a set of cluster labels.But how did we find the optimal number of clusters? The answer is elbow method.We can see the following graphs what are the optimal cluster range to consider so as to get optimal results:









After finding the optimal number of clusters we will try to identify ,to which cluster shall we assign each of our test data points by which the distance between that point and that centroid cluster.In the later part of this report I have discussed the rmse value,its scatter plots,and the clusters.

2.Agglomerative Clustering:

This is a bottom-up approach where each data point starts as its own cluster and clusters are iteratively merged together based on a similarity measure until all data points belong to a single cluster or until a stopping criterion is met.The Dendogram can help us tell what are the best possible number of clusters.

In our project we are using Scikit's inbuilt function to train our model and hence the optimal number of clusters synonymous with it. The results are similarly discussed at the end as we shall for the rest of our algorithms.

3.DBSCAN Algorithm:

The DBSCAN algorithm works by grouping together closely packed points based on two parameters:

1. Epsilon (ε): This parameter defines the radius within which to search for neighboring points. Points within this radius are considered to be part of the same cluster.
2. MinPts: This parameter specifies the minimum number of points required to form a dense region (i.e., a cluster). Points that have at least MinPts neighbors within the radius of ε are considered core points.

This algorithm is highly sensitive to noise present in the dataset.When we ran our algorithm and tried to find clusters on the basis of distance matrix we have found that around 1540 of the data out of 2200 points were marked -1 , i.e. they were considered as noise points and the remaining were formed into 0 numbered clusters and a few to 1 numbered.Thus the results we were getting were completely off from the results we were expecting hence we decided not to proceed with this algorithm. This again reiterates our facts that there is not always one good algorithm for all types of datasets.

4.t-sne:

This is the last of our algorithms we have applied for finding clusters within our data. This is more of a dimensionality reduction technique rather than a clustering algorithm.We have tried to do the same here , we have tried to break down our data points such that we have only 2 dimensions and we visualize the pattern of our clusters. Thus once we have obtained this result we may if require proceed with further clustering additions to be utilized on top of this.Thus we believe the running K-means or agglomerative clustering on top of this algorithm will fetch us the best possible results ,else if for just representation we may just rely on the plots to visualize our cluster.

Testing: Sometimes it is not possible to find distance between the cluster centroid and the test data point thus w e find the similarity between each of the data points with the test data point and then average it out . Then within the cluster we find linear regression between similarity of molecules and the melting point of the molecule. Thus we end up with rmse as our accuracy measure.But blindly finding out rmse does not lead to any meaningful sense thus we also normalized the rmse value with max-min value to give it a better meaning ,which can better explain our model’s results.

Output:

K-Means: Firstly let us see how k-mean performs with finding our solution

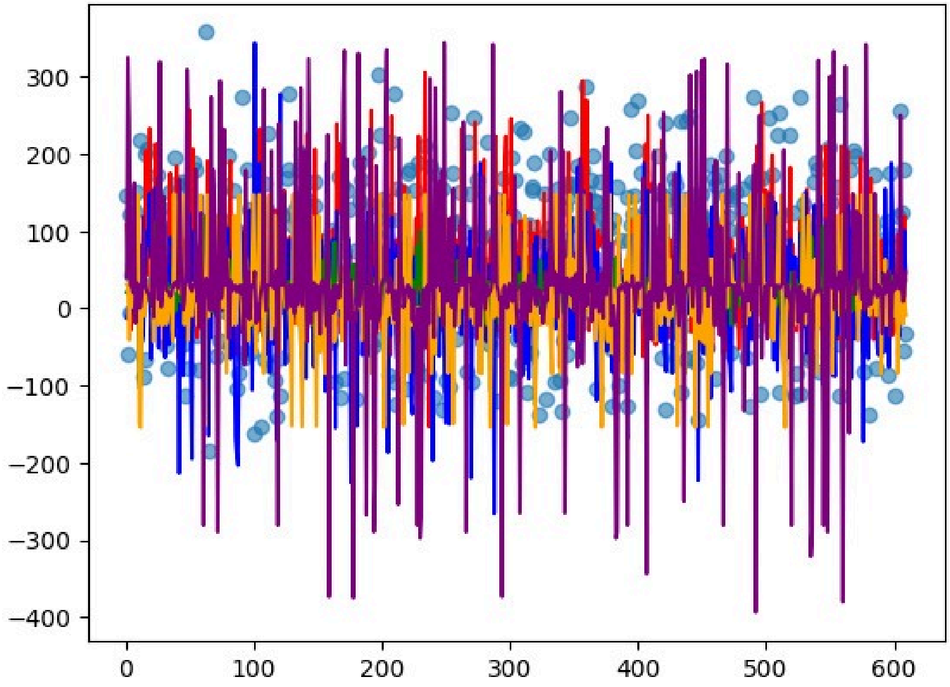


Fig. Predicted melting point vs molecules

There are some extreme values as we can see from the figure which are mostly the outliers let us see another result .

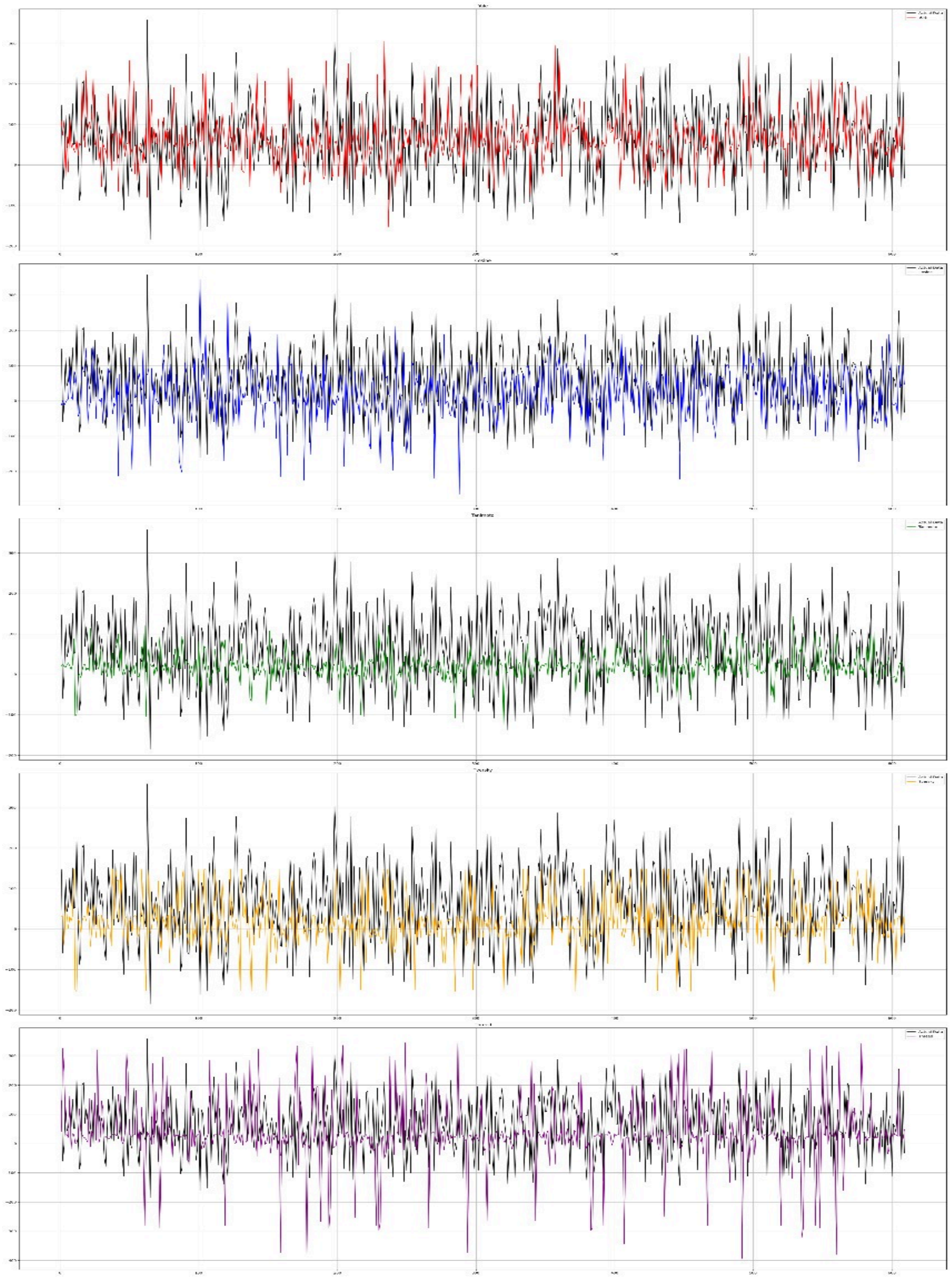


Fig. comparison of predicted and actual melting point vs molecule(for different similarity measure)

Hierarchical Clustering(Agglomerative):

Let us see how many cluster numbers we get for different similarity measures:

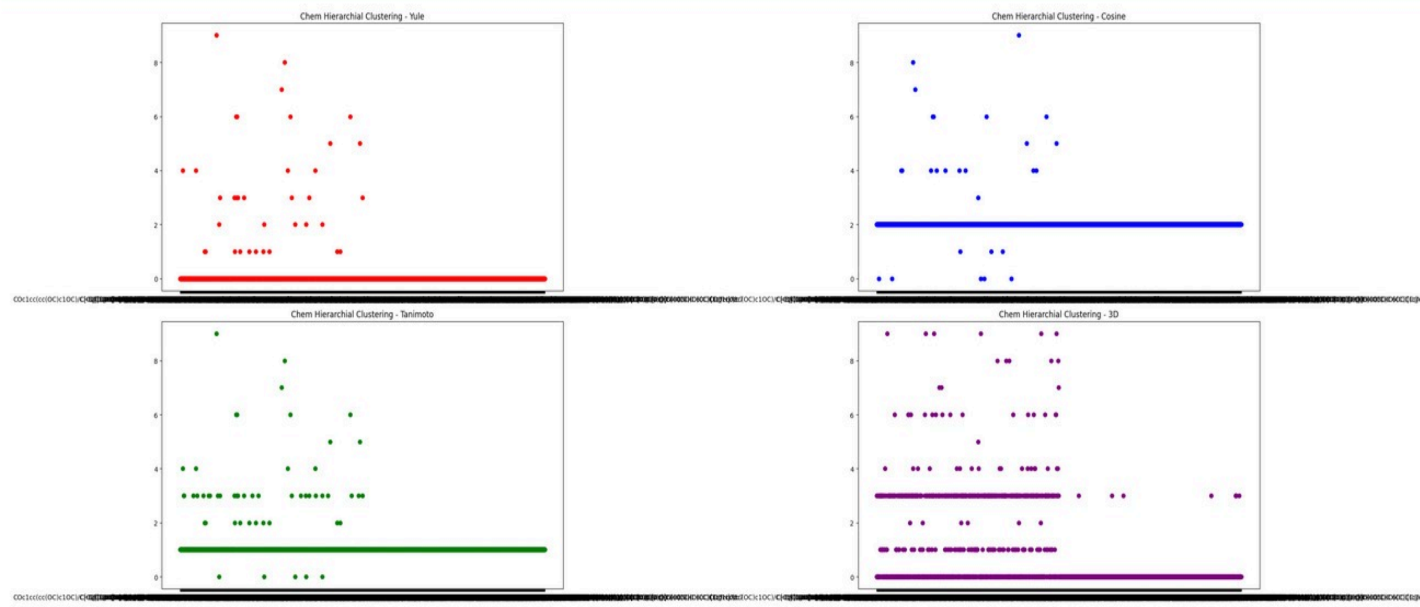


fig .cluster number vs molecules

The other important result:

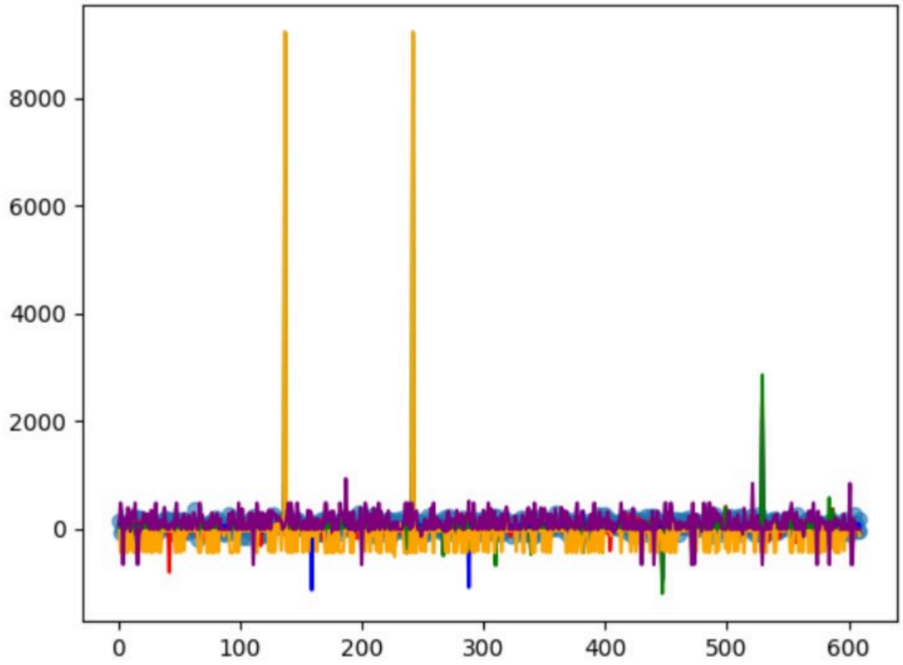


fig . Predicted melting point for each molecule

The next graph tell us the plot of rmse and plots actual vs predicted to visualize how our algorithm does.

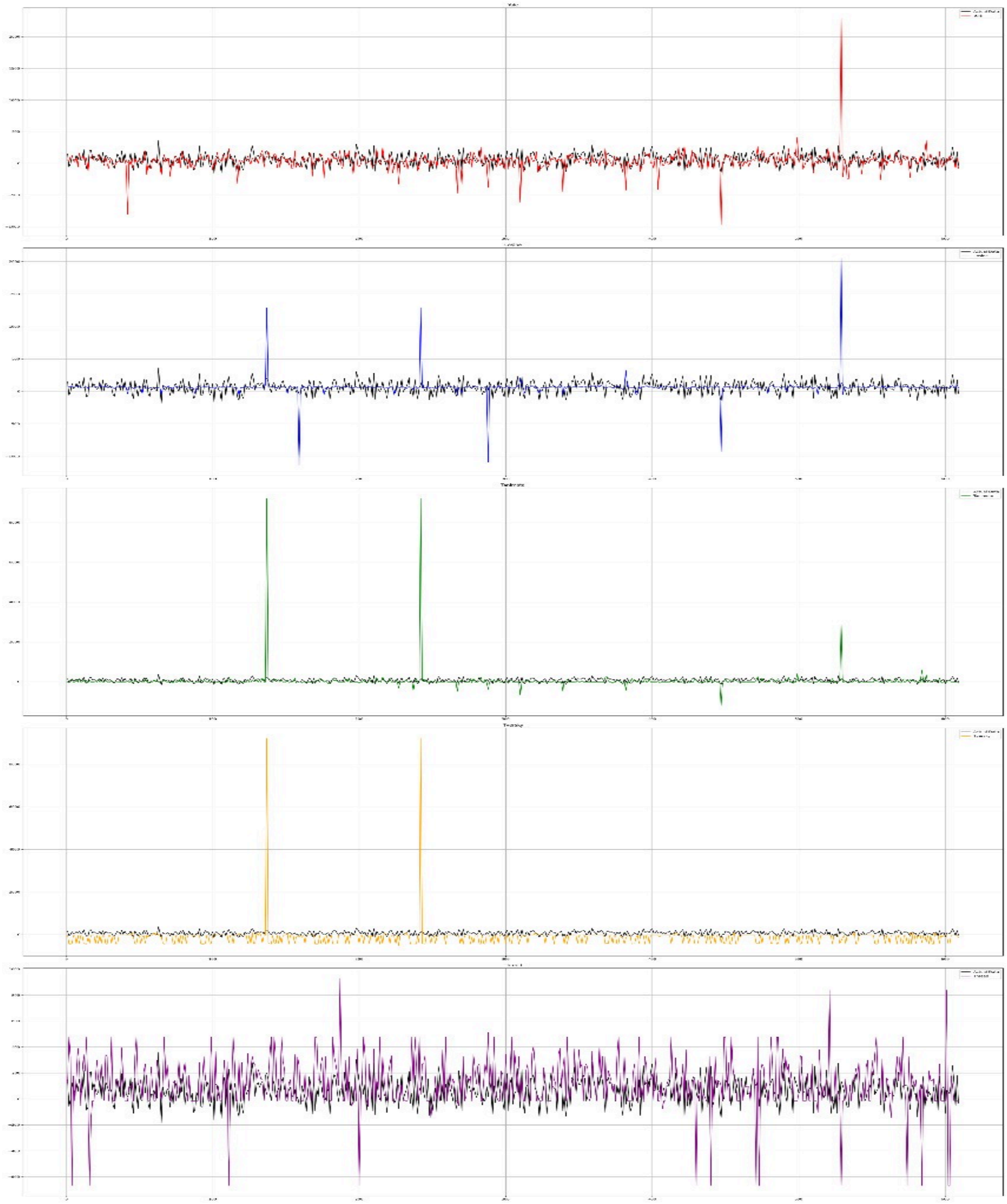


Fig. Comparison of predicted and actual mps vs. molecule w.r.t different similarity measures

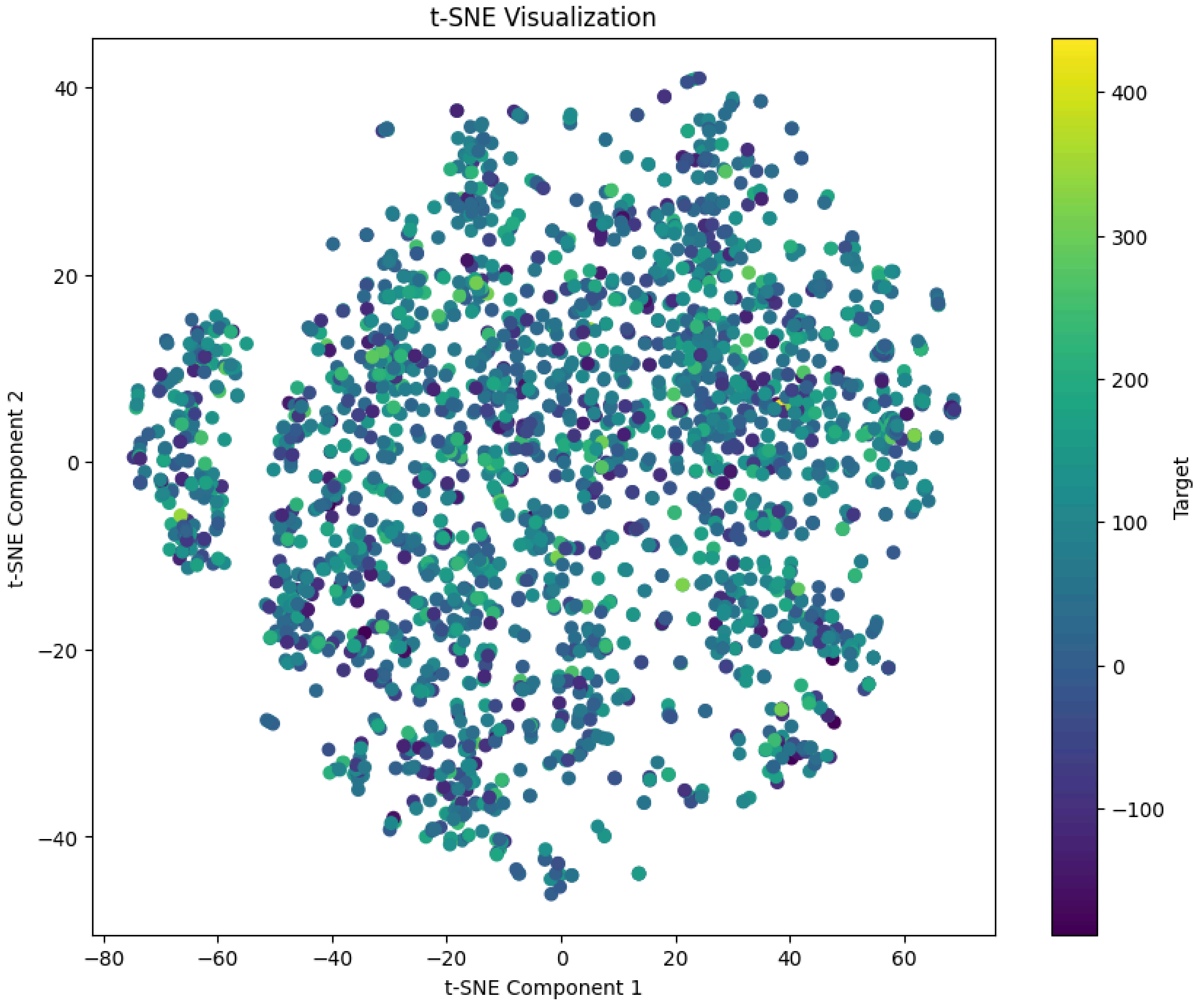
DBSCAN:

As mentioned above we have not proceeded with this algorithm after we have derived with cluster labels for our training data as most of the data was being classified as noise.DBSCAN is classifying our data points that don't have enough neighbors within a specific radius or lack the minimum number of neighbors to form a dense region as noise ,hence the huge mis clustering.

T-sne:

While t-SNE itself does not perform clustering, it can be used as a preprocessing step before applying clustering algorithms to identify clusters in the lower-dimensional space. After transforming the data using t-SNE, you can then use clustering algorithms like K-means, DBSCAN, or hierarchical clustering to identify and label clusters in the embedded space.

We have used this algorithm and tried to visualize our data on a 2d spectrum the following results were produced when we ran our data on this model:


t-SNE Visualization
t-SNE Component 1
t-SNE Component 2
Target