Property Predictor

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

(No physical Chemistry calculations involved.)

Submitted By:

(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:

Index	Paper Name	Paper Description	Paper Techniques Used
1.	Graph-Based Clustering of Chemical Compounds for Property Prediction	This paper proposes a novel graph-based approach for clustering chemical compounds according to their molecular structures to improve property prediction accuracy. The method involves representing chemical compounds as graphs and employing graph clustering algorithms to identify structurally similar compounds. Evaluation on diverse datasets demonstrates the effectiveness of the approach in enhancing property prediction models compared to traditional methods. The study contributes to advancing predictive modeling in chemical informatics by leveraging molecular structure information for clustering and prediction tasks.	Molecular representations such as Smiles ,Selfies have been used while data creation.

2.	Predicting Physical Properties of Molecules Using Machine Learning Models	This paper presents a comprehensive study on utilizing machine learning models to predict physical properties of molecules based on their structural features. Various machine learning algorithms, including deep learning models, are employed to analyze molecular structures and predict properties such as solubility, boiling point, and surface tension. The study investigates the impact of different feature representations, model architectures, and training strategies on prediction performance.	Some of the datasets have been used from the paper.
3.	A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise	This paper introduces the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm, which is designed to identify clusters of varying shapes and sizes in spatial databases with noise. Unlike traditional distance-based clustering algorithms, DBSCAN doesn't require a predetermined number of clusters and can handle outliers effectively. Instead, it relies on the concepts of density reachability and density connectivity to define clusters based on regions of high data density.	The algorithm modeling data and the algorithm for testing DBSCAN has been derived from this paper.
4.	Visualizing Data using t-SNE	This paper introduces t-Distributed Stochastic Neighbor Embedding (t-SNE), a technique for dimensionality	T-sne algorithms,labeling and visualization of clusters has been possible due to this paper.

		reduction that is particularly effective for visualizing high-dimensional data in a low-dimensional space. t-SNE aims to preserve local and global structure in the data, making it well-suited for visualizing clusters and patterns. The paper provides a detailed description of the t-SNE algorithm and demonstrates its effectiveness through various experiments on real-world datasets. It has since become a widely used tool for exploratory data analysis and visualization in machine learning and data science.	
5.	Clustering of small molecules: new perspectives and their impact on natural product lead discovery	In this paper, the clustering of small molecules takes center stage, showcasing its vital importance in computer-aided drug discovery and virtual screening applications. We delve into the structuring of chemical data within appropriate subspaces of the chemical space, emphasizing its role in representative dataset sampling and the generation of tailored libraries with diverse chemical coverage.	Clustering principles and representation techniques.
6.	Prediction of small-molecule compound solubility in organic solvents by	Rapid solvent selection is crucial in chemistry, yet accurate solubility prediction remains challenging. In this study, we developed machine learning models to predict compound solubility in organic solvents. A dataset of	Experimental dataset containing smiles which was used for training our molecules to convert it to smiles format.

	machine learning algorithms	5081 experimental temperature and solubility data points in organic solvents was collected and standardized. Molecular fingerprints were chosen to characterize structural features.	
7.	Hierarchical Clustering for Property Graph Schema Discovery	In this paper, we explore the application of hierarchical clustering techniques for discovering property graph schemas that encode essential structural and functional attributes of molecules. Building upon this schema discovery, we develop predictive models for molecular properties using machine learning approaches. Our methodology is validated on a diverse dataset of molecular structures and their associated properties, showcasing the efficacy of hierarchical clustering in uncovering informative patterns.	Graph based similarity between molecules. And Hierarchical Clustering.

8.	Machine Learning Approach to Predict Physical Properties of Polypropylene Composites: Application of MLR, DNN, and Random Forest to Industrial Data	In this paper, a machine learning approach is explored for predicting physical properties of polypropylene composites through the application of multiple algorithms to industrial data. Specifically, the efficacy of multiple regression (MLR), deep neural networks (DNN), and Random Forest.This study provides insights into the suitability of different machine learning techniques for predicting physical properties in industrial settings, offering valuable guidance for optimizing composite materials design and manufacturing processes.	Multiple linear regression utilized after clustering.
9.	DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN	The recently published paper presents an interesting complexity analysis of the DBSCAN problem with Euclidean distance	Different distance metrics have been utilized with the help of this paper
10.	t-SNE-CUDA: GPU-Accelerated t-Distributed Stochastic Neighbor Embedding	T-SNE-CUDA is introduced, a GPU-accelerated implementation of t-distributed Symmetric Neighbour Embedding for visualizing datasets and models and significantly outperforms current implementations with 50-700x speedups on the CIFAR-10 and MNIST datasets.	Rapids implementation of t-sne algorithm used for comparative purposes.

<u>Challenges faced and handling techniques employed</u>:

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. [1]
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 [7] [5], K-Means, DBSCAN, and t-SNE, to identify clusters of similar molecules based on their properties.

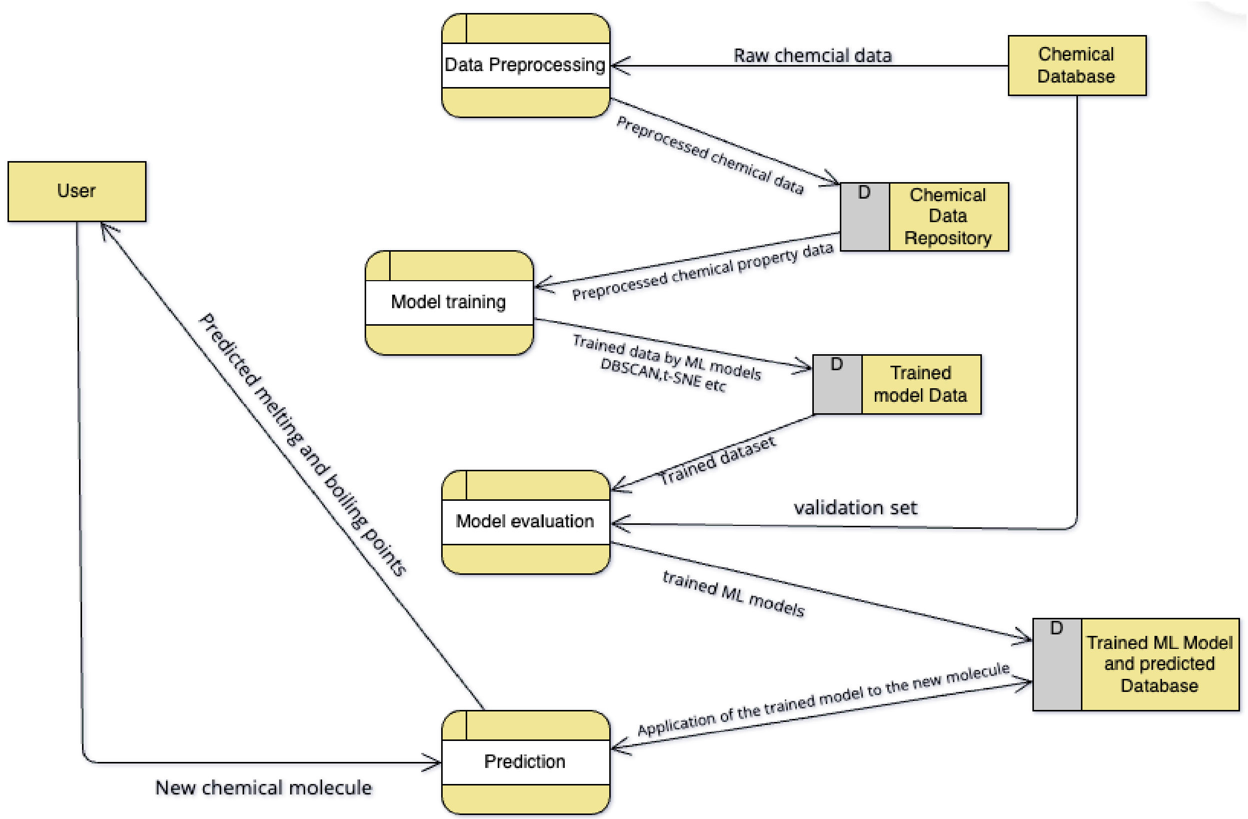
2. **Optimization of K in K-Means Clustering:** Determining the optimal number of clusters (k) in K-Means clustering is crucial but challenging. We utilized the elbow method to optimize k, ranging from 5 to 20 clusters, by evaluating the inertia for different values of k.
3. **Similarity Metrics Selection:** 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.[6]
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. [4]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
Preprocessed chemical data
User
D
Chemical Data Repository
Model training
Preprocessed chemical property data
Trained data by ML models DBSCAN,t-SNE etc
D
Trained model Data
Predicted melting and boiling points
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. [2]

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. [1]

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) [2] 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.[9]

2.Cosine Distance: Cosine Distance is but 1-Cosine similarity where we can calculate cosine similarity as:
Cosine Similarity(Cs)= x.y/√x.x√y.y where '.' is the dot product between the two vectors[9]

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:

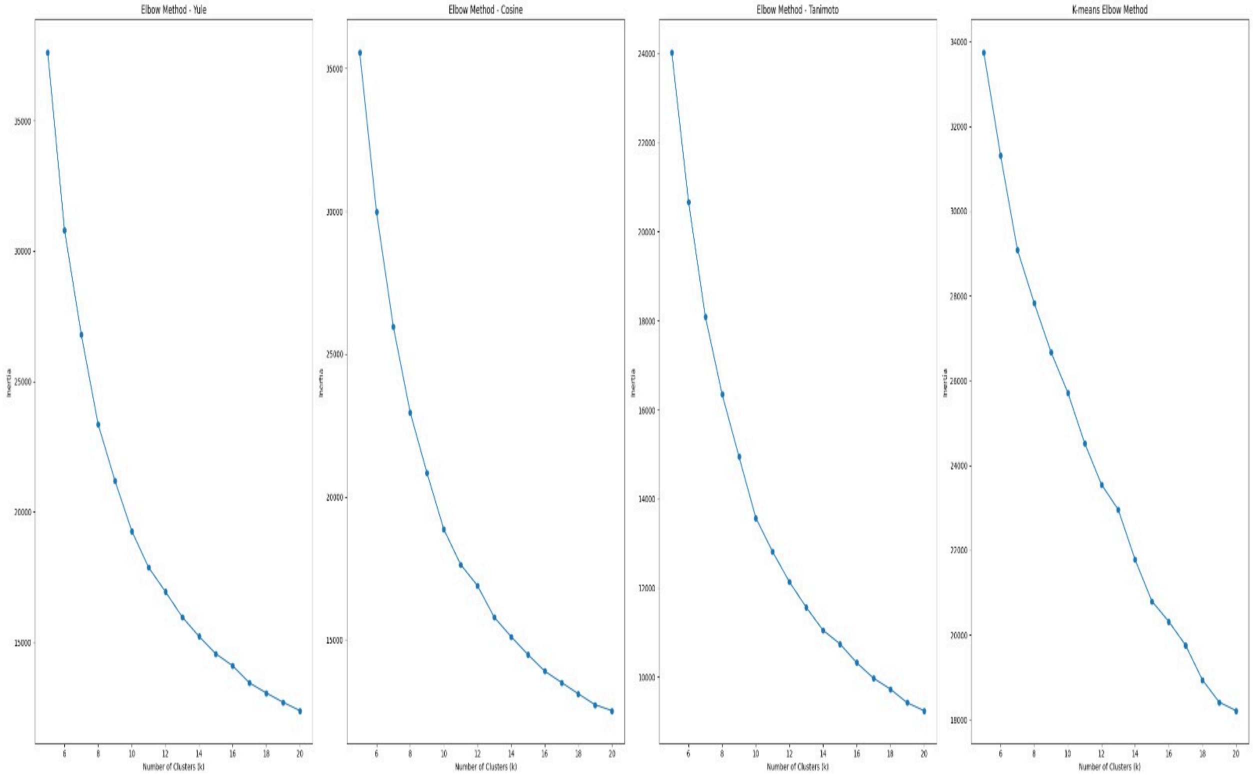


Fig-1: Elbow method to find optimal k-value in k-means.

Fig-1 description: Elbow method allowed us to tune the k(number of clusters) parameter for k-means clustering. K value was tuned in the range 5-20.

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.[7][5]

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. [4] 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.[10]

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.

Results and Analysis:

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

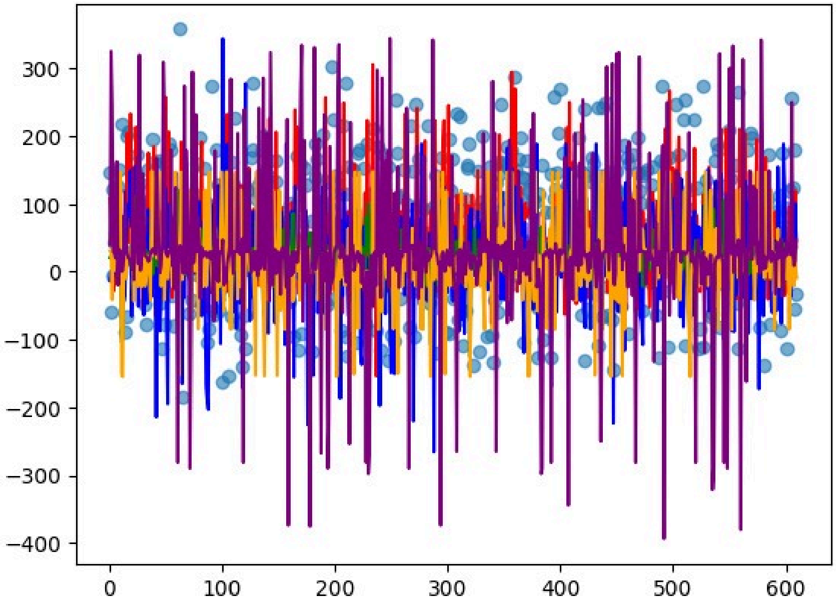


Fig-2. Predicted melting point vs molecules

Fig-2 Description: There are some extreme values as we can see from the figure which are mostly the outliers let us see another result .

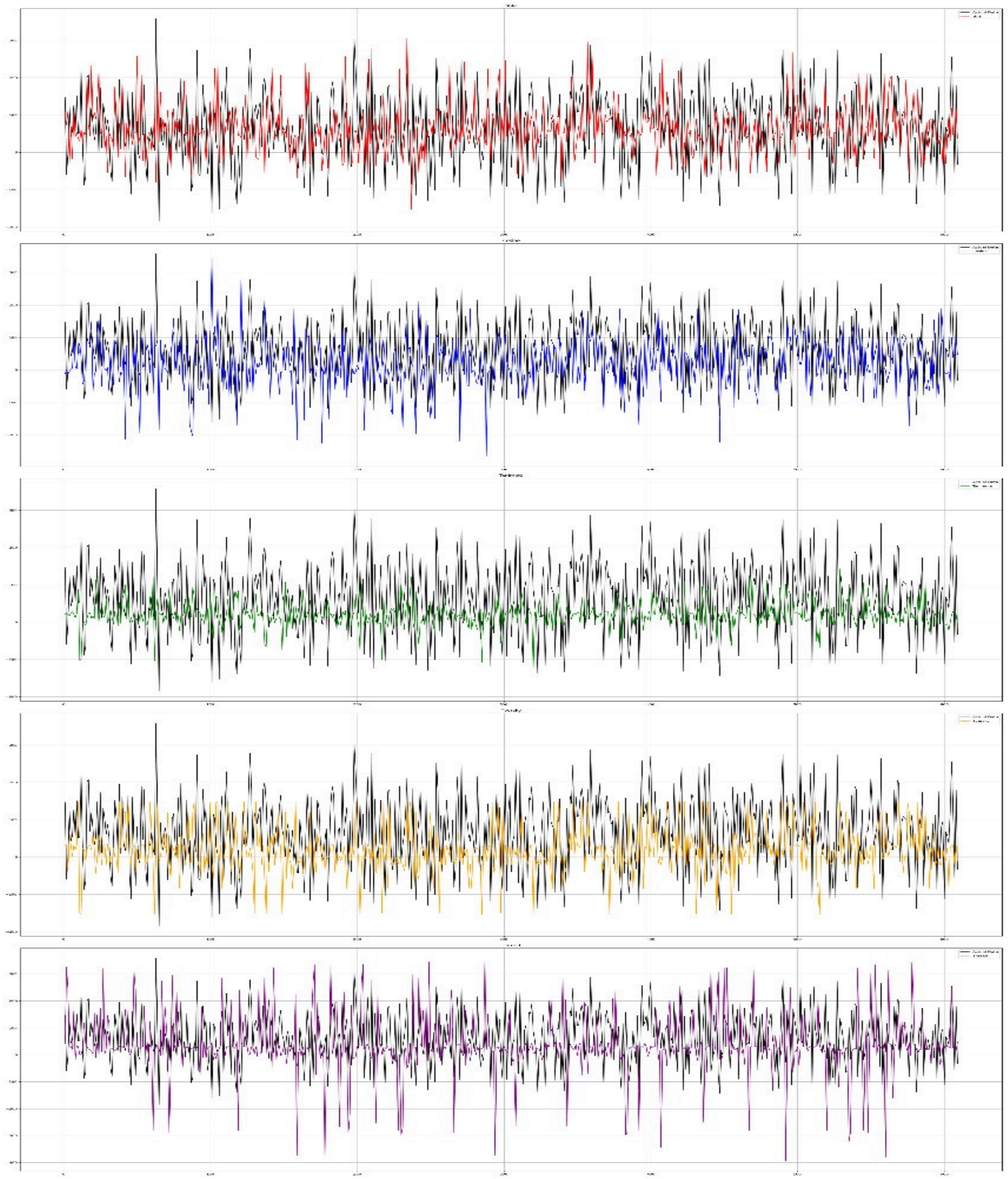


Fig-3. comparison of predicted and actual melting point vs molecule(for different similarity measure)

Fig-3 Description: The comparison allows us to understand that the distribution of melting point predicted vs. real is similar.

Hierarchical Clustering(Agglomerative):
Let us see how many cluster numbers we get for different similarity measures:

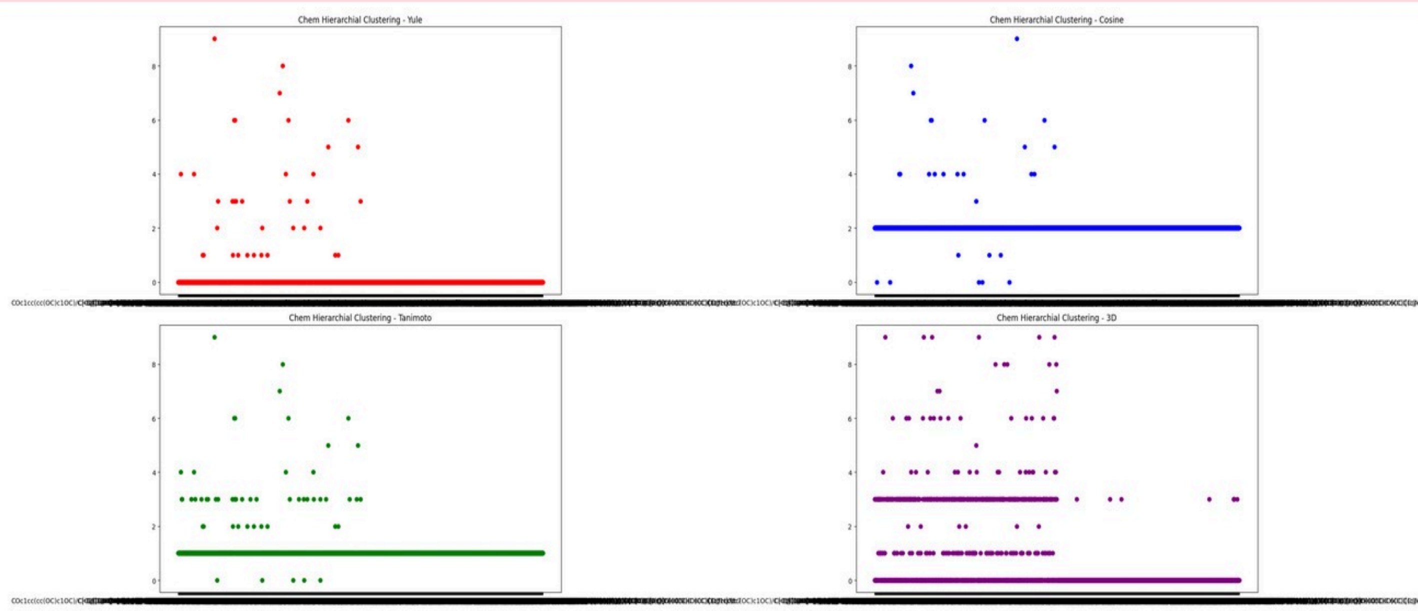


fig-4.cluster number vs molecules

Fig-4 Description: The figure shows the agglomerative clustering and its cluster labels.

The other important result:

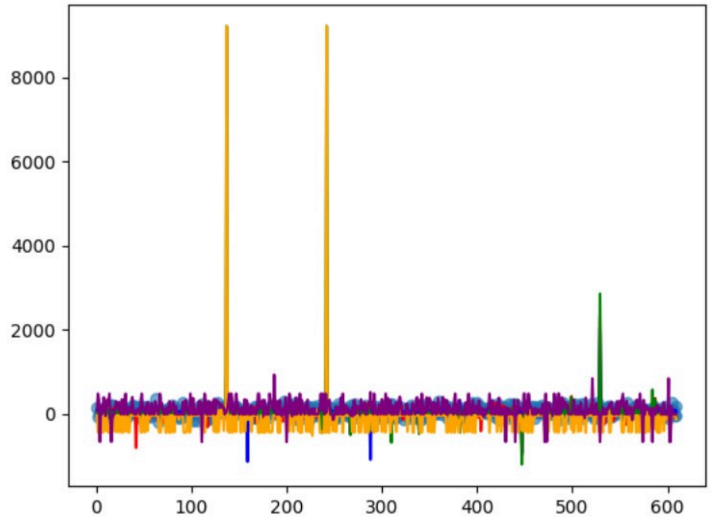


fig-5 . Predicted melting point for each molecule

Fig-5 Description: The comparison allows us to understand that the distribution of melting point predicted vs. real is similar.

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

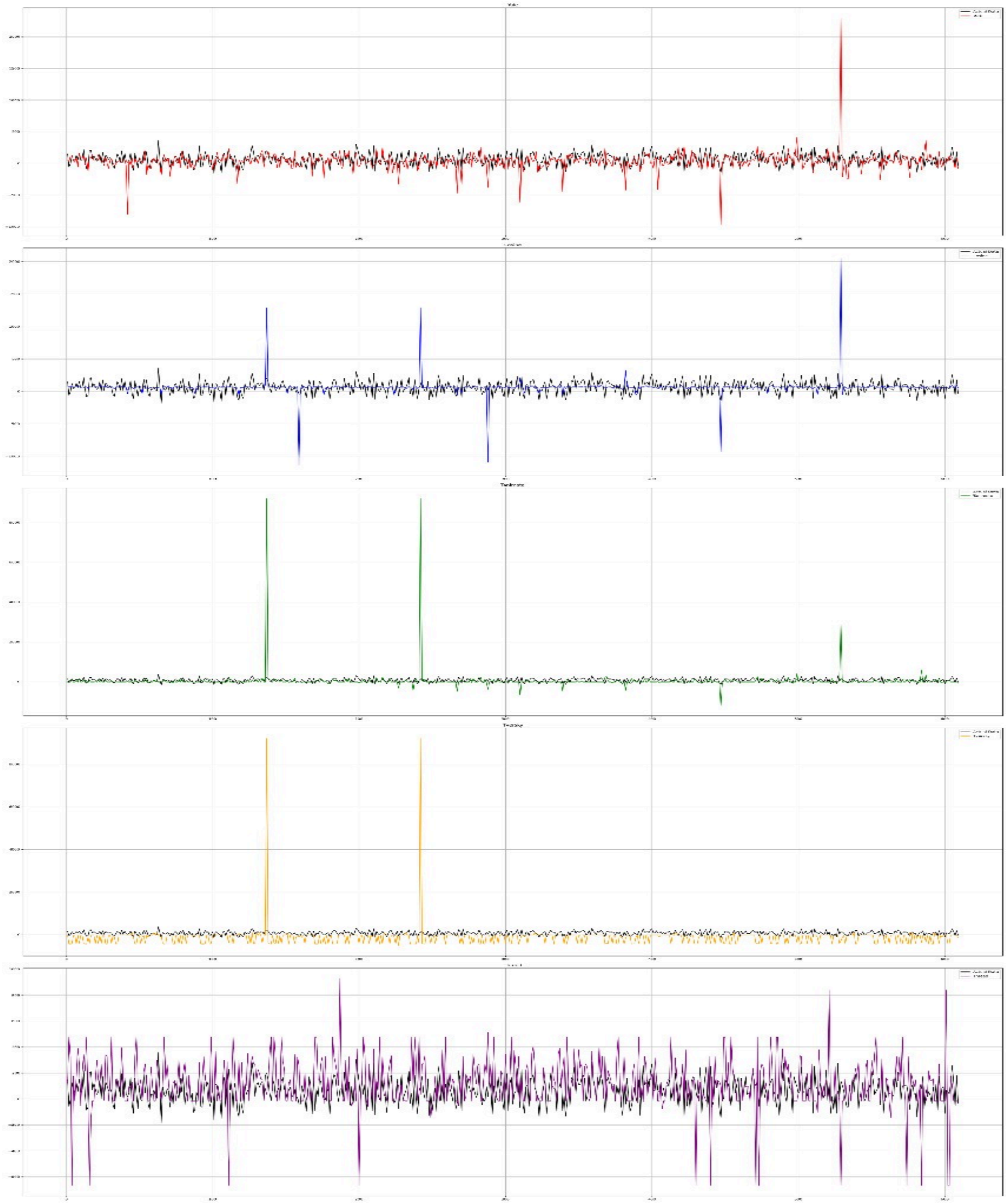


Fig-6. Comparison of predicted and actual mps vs. molecule w.r.t different similarity measures

Fig-6 Description: The figure shows that using tanimoto similarity measure provides us with the better distribution of melting points w.r.t the actual distribution.

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:

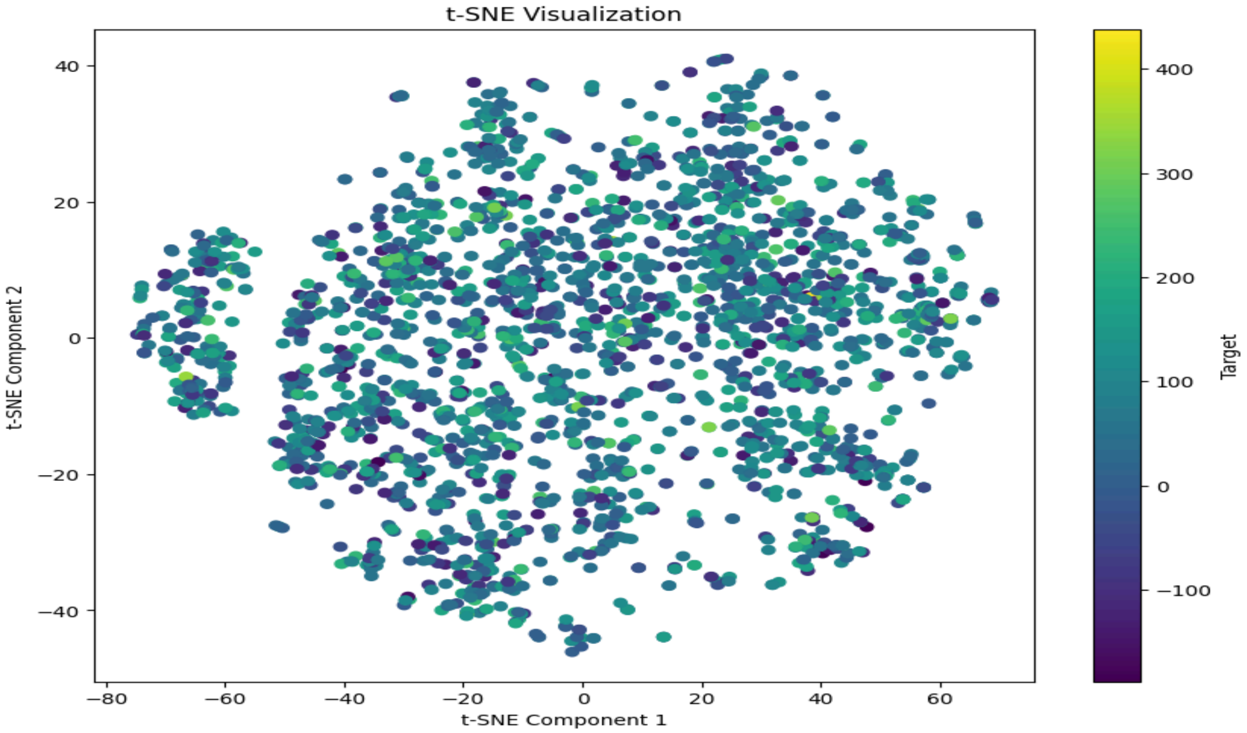


Fig-7: t-SNE plot.

Conclusion:

Metrics	Value
Rmse yule	0.37584796416394733
Rmse cosine	0.25526376099913534
Rmse tanimoto	0.1833699117500185
Rmse tversky	0.2445295732473339
Rmse threed	0.32283286198035377

Results-table-1: K-means

Metrics	Value
Rmse yule	0.28226432004111884
Rmse cosine	0.2806375375089301
Rmse tanimoto	1.0101861888416486
Rmse tversky	1.1074534199871338
Rmse threed	0.3866809051775003

Results-table-2: Agglomerative Clustering.

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.

T-sne:
The visualization in the above graph shows good separation between clusters.tsne requires manual clustering using the cluster labels labeled in the visualization plot.Automated clustering using these labels would be one of the future scopes.

We can conclude from the above data that utilizing K-Means with tanimato similarity measure has resulted in the best accuracy. Using unsupervised learning techniques, provides an edge in ease of computation and labeling. So this project is well-suited for the problem statement.

Future Scope:

We have tested our data using multiple unsupervised clustering models.We hope to further expand it to multiple algorithms and see which algorithm or combination of algorithms produces the best results.

We shall also further work on DBSCAN and see if any noise removal algorithms would do any good and or any combination of algorithms works.

Currently we have implemented our code to test it on melting and boiling point ,we further hope to generalize it to many more properties such as the structure of the molecule,its solubility etc.

We also plan to release an application utilizing the same for future applications and for user -friendly model testing.

References:

1.Graph-Based Clustering of Chemical Compounds for Property Prediction:
https://www.nature.com/articles/s42004-024-01155-w
2.Predicting Physical Properties of Molecules Using Machine Learning Models:
https://link.springer.com/article/10.1007/s00521-021-05961-4
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.Clustering of small molecules: new perspectives and their impact on natural product lead discovery:
https://pubmed.ncbi.nlm.nih.gov/17125183/
6. Hierarchical Clustering Schemes:
https://jcheminf.biomedcentral.com/articles/10.1186/s13321-021-00575-3
7.Hierarchical Clustering for Property Graph Schema Discovery:
https://openproceedings.org/2022/conf/edbt/paper-139.pdf
8. Machine Learning Models for Predicting Physical Properties of Organic Compounds:
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9459971/
9. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN:
https://dl.acm.org/doi/10.1145/3068335
10. t-SNE-CUDA: GPU-Accelerated t-Distributed Stochastic Neighbor Embedding:
https://www.semanticscholar.org/paper/T-SNE-CUDA%3A-GPU-Accelerated-T-SNE-and-its-to-Modern-Chan-Rao/