Norwegian University
of Life Sciences

Master’s Thesis 2023 30 ECTS

Faculty of Science and Technology
Professor Kristian Berland

Prediction of Melting Temperature of Organic Molecules using Machine Learning

Aditya Dey

MSc Data Science

The page is intentionally left blank.

Acknowledgments

I would like to express my deep appreciation to my supervisor, Prof. Kristian Berland, for his exceptional guidance, encouragement, and support throughout my research. His invaluable insights, constructive feedback, and unwavering commitment have been instrumental in shaping this project.

I would also like to extend my sincere gratitude to Seyedmojtaba Seyedraoufi and Elin Sødahl for their invaluable assistance, generous support, and insightful guidance. Their contributions have been immensely helpful in advancing my research and achieving my goals.

Finally, I would like to thank my family and friends for their unconditional love, unwavering support, and understanding throughout this journey. Their constant encouragement and belief in me have kept me motivated and inspired me to pursue my passion.

Aditya Dey
Ås, May 15th 2023

The page is intentionally left blank.

Abstract

Accurate prediction of the melting point of oral drugs is crucial for understanding their chemical properties. Early identification of these properties aids in the screening of potential drugs, thereby saving resources in the pharmaceutical industry's discovery and manufacturing processes. The prediction of organic molecules is a complex task due to many factors that affect entropy and enthalpy forces within a molecule, which are dependent on various factors like shape, electronegativity, flexibility, rotatability, intermolecular bonding, etc.

In this study, we curated a combined dataset of organic molecules, extracted from the Open Notebook Science Dataset and Cambridge Structure Database. The dataset consists of molecules composed of carbon, oxygen, nitrogen, sulfur, phosphorous, and halogens, exhibiting a wide range of melting point temperatures and molecules with complex structures. To gain insights into the significance of each feature and its contribution to melting point prediction, we divided the combined dataset into four subsets based on the number of bonds an atom can form.

We perform feature engineering on these datasets by studying the physical and chemical properties known to impact melting points. Numerical features were derived from the molecules, capturing relevant information. Additionally, we utilized embedding features without any modifications.

Machine learning models were trained using both numerical and embedding features, with the accuracy evaluated through R^2 scores and root mean squared error values. We set the model trained on embedding features as a benchmark for our model and features to surpass. Our machine learning models exhibited good performance, outperforming the benchmark and achieving good prediction accuracy.

Furthermore, we conducted an in-depth analysis of the results to assess the impact of individual features on the models. We observed physical shape features and the presence of specific substructural groups exhibited a strong correlation with melting point prediction. To explore the relationship between features, we performed a principal component analysis.

The findings of this study have important implications for drug development, formulation, and optimization of manufacturing processes. Accurate prediction of melting points enhances drug screening procedures and aids in the design of effective pharmaceutical products.

The codes are available in github.

The page is intentionally left blank.

Contents

The page is intentionally left blank.

List of Figures

List of Tables

Abbreviations

logP	log octanol-water partition coefficient
vdW	Van der Waals
AdaBoost	Adaptive Boosting
AI	Artificial Intelligence
ANN	Artificial Neural Network
CIF	Crystallographic Information File
CSD	Cambridge Structure Database
DT	Decision Tree
GSE	General Solubility Equation
ML	Machine Learning
MPbAP	Melting point based Adsorption potential
MSE	Mean Squared Error
NLP	Natural Language Processing
ONS	Open Notebook Science
PCA	Principal Component Analysis
QSPR	Quantity Structure Property Relationship
R^2	Coefficient of determination
ReLU	Rectified Linear Unit
RFE	Recursive Feature Elimination
RBF	Radial Bias Function
RMSE	Root Mean Squared Error
SDF	Structure Data File
SMILES	Simplified Molecular Input Line Entry System
SSE	Sum of Squared Error
SST	Total sum of squares
SVM	Support Vector Machines

SVR	Support Vector Regressor
XGBoost	Extreme Gradient Boosting

The page is intentionally left blank.

1 Introduction

Melting point is an important physical property in the pharmaceutical industry in the process of creating and manufacturing oral drugs. Oral dosage is a common method of drug delivery route due to patient adherence, cost-effectiveness in manufacturing, ease of taking the drug, and flexibility in the designing of the dosage. The intended effect of taking orally is to have the drug reach different parts of the body via the bloodstream. The effectiveness of the drug depends on the adsorption, absorption, and solubility in the digestive system where the drug is broken down by enzymes.

The melting point has a relationship with Adsorption which refers to the capability of the drug to attach or bound to the surface of a material(enzyme) which can then transport the drug to the target site in the body. Melting point based Adsorption potential (MPbAP) as shown in equation (1) states that higher adsorption is achieved for any given dose at lower melting points and decreases with increasing melting points [8]. It means a drug with a dose of 100 mg and a melting point of 200°C will have better adsorption compared to a drug with the same dose but a higher melting point of 300°C. If the dose is increased from 100 mg to 200 mg then the adsorption rate will still be better for drugs with lower melting points. This information is required during oral drug selection of thousands of compounds, to predict a drug's intestinal adsorption rate allowing for rapid and efficient drug selection. But the selection process based on MPbAP depends on identifying the melting point of the organic compound which then allows calculation of the adsorption rate and the dose of the drug that can be administered.

$$[0.5 - 0.01(\mathrm{MP} - 25)] - \log(4 \times \mathrm{Dose}) \geq 0 \tag{1}$$

Melting point also has a relationship with drug solubility, which is the ability of a drug compound to dissolve in liquid to form a homogeneous solution. A drug compound that has low solubility when administered to a patient will have poor bioavailability leading to lower potency [9] and will require high-dose administration. This will instigate other undesirable characteristics like nausea, vomiting, and abdominal pain. The aqueous solubility($\log S_w$) of a drug is given by General Solubility Equation (GSE) as shown in equation (2) which shows the mathematical relationship to the log octanol-water partition coefficient (logP) and its melting point [10]. A compound with a higher melting point tends to have stronger intermolecular forces that are less resilient to breaking. These are the same intermolecular forces that are to be overcome when the compound is consumed orally with water. So a drug compound with a higher melting point will have stronger intermolecular forces that can lead to lower aqueous solubility. Therefore melting point has a strong relationship with solubility and identifying drugs with lower melting points can assist the drug discovery and screening process.

$$\log S_w = 0.5 - 0.01(\mathrm{MP} - 25) - \log P \tag{2}$$

Thus a crucial part of drug discovery is to identify the melting point that determines a drug's profile. This profile is called ADMET - adsorption, distribution, elimination, and toxicity. The ADMET profile of a drug if known from the beginning allows better investment of resources and saves time in preclinical and clinical testing. These ADMET properties as discussed above are dependent on melting point and therefore focusing the resources on attempting to predict the melting point of the drug in advance can highlight a drug's potential during the drug discovery and selection process.

The melting point is also an indicator of purity of a compound and the process of identifying the purity is called purity determination. Each compound has a different melting point and a pure compound will sharp melting range that is less than 5°C [11]. When the compound has impurity it has a broad melting range greater than 5°C [11]. increases therefore allowing us to identify the purity of a compound. Impurities can deter the product's quality in the pharmaceutical industry.

Producing a compound involves multiple stages that begin with the sourcing of material, synthesis of the compound in the laboratory, large-scale manufacturing, storage, and delivery. Along these stages, the compound is exposed to variations in temperature and pressure and other substances which can lead to the introduction of impurities in the main compound. For example, if the quality of the starting materials, reagents, and solvents is impure in nature, then the final compound's quality dramatically changes [12]. Also after the synthesis residual solvents if still present in the main compound will also add to its impurity. Thus purity determination by identifying the melting point of the manufactured product and validating it with the true melting point of the product during laboratory synthesis acts as a quality control check during each stage of the manufacturing process in pharmaceutical industries.

This shows the importance of the melting point is not bound alone to drug discovery but every step that begins from discovery leading to the selection, pre-clinical testing, clinical testing, manufacturing, and its effects on the patients after prolonged usage. Thus our goal should be to attempt to identify the melting point of a compound to eliminate or select them in the early stages of screening.

What is Melting Point?

According to the law of thermodynamics, the Gibbs free energy ΔG is determined by the change in enthalpy(ΔH) and entropy(ΔS) of a reaction as shown in equation (3). When $\Delta G = 0$ then $T = T_m$ is given by the equation (4) where melting point temperature is the ratio of enthalpy and entropy.

$$\Delta G = \Delta H - T\Delta S \tag{3}$$

$$T_m = \frac{\Delta H_m}{\Delta S_m} \tag{4}$$

Enthalpy of melting is the amount of heat required to melt 1 mole of a substance at constant pressure and temperature to change the phase from solid to liquid. In a crystal lattice, the neighboring molecules are held together by intermolecular forces restricting their movement. The heat supplied to the substance breaks these intermolecular forces allowing the molecule to move freely and thus begin the transition to a liquid state. Therefore enthalpy of melting is associated with the amount of heat required to break all intermolecular forces within 1 mole of a substance.

The intermolecular forces are Van der Waals (vdW) forces and hydrogen bonds. vdW are weak intermolecular forces that occur in molecules caused by temporary shifts in electron distribution within the molecule creating a partial negative and positive charge that can attract atoms from nearby molecules. They include dipole-dipole, dipole-induced dipole, and London forces [13]. Hydrogen bonds are stronger than vdW's forces and are formed between an electronegative atom and a hydrogen atom [13] where it is covalently bonded to one electronegative atom within the molecule and an electrostatic force with another electronegative atom from a different molecule.

The force contribution of these atoms and groups in a molecular crystal towards enthalpy of melting is given by the equation (5) where n_i is the number of group i in the molecule and m_i is the contribution of each group i to the enthalpy of melting [14].

$$\Delta H_m = \sum_i n_i m_i \tag{5}$$

The entropy measures the degree of disorder or randomness in the molecules in the current phase of the thermodynamic system. In the solid phase, the molecules have less degree of freedom to rotation, expansion, and conformation as compared to a liquid phase [15]. The entropy of melting can be described by the Boltzmann relationship equation (6) [14] where R is the Boltzmann Constant with a value of approximately 8.31 J mol^{-1} K^{-1} and p_m is the ratio of probabilities of the number of ways 1 mol of material can exist within the confines of the crystal and liquid given by $\frac{\Omega^\mathrm{C}}{\Omega^\mathrm{L}}$.

$$\Delta S_\mathrm{m} = -R \times \ln p_\mathrm{m} \tag{6}$$

This entropy can be further explained using Carnelley's Rule of Symmetry which states that among the isomers of a molecule, the isomer with the highest rotational symmetry number shall have a higher melting point [14]. Rotation symmetry number(σ) refers to the number of identical images a molecule can make when rotated within 360° in any direction. Carnelley also states that molecules that have long chains tend to have a low melting point due to higher flexibility that creates many conformations. It is given by flexibility number(ϕ) that relates to conformations possible in crystal to that of liquid. The compactness of a molecule refers to the arrangement of atoms in a molecule that are closer to each other and is defined using eccentricity number(ϵ). Eccentricity identifies the flatness of a molecule.

Thus according to Carnelley's principle, we can elaborate the entropy equation (6) into equation (7) [14] which says the total entropy of melting can be described using rotation symmetry number(σ), flexibility number(ϕ) and eccentricity number(ϵ).

$$\Delta S_m = \text{Const} - R\ln\sigma - R\ln\phi - R\ln\epsilon \tag{7}$$

Hence, our aim now is to understand and describe the factors determining the shape and intermolecular forces of a molecule that can assist to understand the effects of entropy and enthalpy allowing the prediction of melting points.

Prediction of melting point is usually performed using Quantity Structure Property Relationship (QSPR) which is a regression method where a molecule's physical and chemical properties are used as predictor variables(X) to determine a regression function($f(X)$). This regression function is used to predict the response variable(Y). The regression function can be learned by a Machine Learning (ML) model trained on a dataset of molecules [16].

The predictor variable can be numerical, graphical, or embedded features that can represent the molecule. Numerical features may represent a quantitative feature that may provide a description of the molecule like molecular weight or volume. Graphical features convert the molecule into the node and edge features [17] and Embedding features convert the property of the molecule into an array [16].

Therefore we use the QSPR method to train ML models on features that describe the entropy and enthalpy of a compound and perform prediction of the melting point temperature of various organic compounds.

1.1 Aims of this Master's Thesis

ML has taken over various fields such as image processing, natural language processing, time series analysis, etc which were earlier deemed as complex problems. Yet in the field of chemo-informatics, determining a basic property like melting point has been a difficult problem due to its complexity [18] that arises from various factors like intermolecular forces, shape, chirality, etc.

Over the years, there have been many attempts in solving this problem and few researchers have been able to provide relationships between these factors that allow for predicting the melting point. But these relationships work only in a subset of molecules and there are no generalized melting point prediction models that can predict for all types of molecules. The prediction of melting point is key to understanding other physical and chemical properties which are of importance in the pharmaceutical and chemical industries.

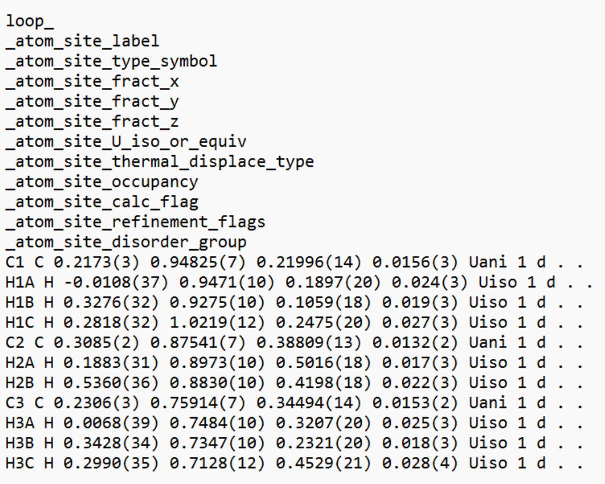
Thus our aim through this research is to perform feature engineering of organic molecules that describes these properties affecting melting points and use ML to analyze and identify its effectiveness to improve the prediction of a melting point. We attempt to build a general purpose ML model that predicts the melting points of a larger subset of organic molecules with a wide temperature range. Along this process, we aim to bridge the gap between chemistry and data science by attempting to build physical and chemical properties into mathematical features that describe the molecule and can be used by the ML model for prediction.

2 Datasets

In this section, we explore various datasets, the elimination methods applied to them, and the necessity to combine them. Further, we will understand the need to segregate our combined dataset into carbon, carbon-halogen, carbon-halogen-oxygen-sulfur, and carbon-halogen-oxygen-sulfur-nitrogen-phosphorous datasets.

2.1 Cambridge Structure Database

The CSD is a comprehensive collection of published organic and metal-organic small molecule crystal structures [19]. The structural information is stored in CIF that comprises chemical formula, volume, cell parameters, atomic coordinates, the method used to extract the information, etc. It is the standard file format specified by the International Union of Crystallography to ensure the build of the structural model is the same in all software [20]. Individual researchers or organizations deposit the CIF into the CSD along with their scientific articles. The CSD Team performs validations and assigns a 'CCDC number' to uniquely identify the crystal structure, which can be directly referred to in other scientific journals using the number. As viewed on March 2023, CSD contains 1,228,093 deposited crystal structures.



```
loop_
_atom_site_label
_atom_site_type_symbol
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_U_iso_or_equiv
_atom_site_thermal_displace_type
_atom_site_occupancy
_atom_site_calc_flag
_atom_site_refinement_flags
_atom_site_disorder_group
C1 C 0.2173(3) 0.94825(7) 0.21996(14) 0.0156(3) Uani 1 d . .
H1A H -0.0108(37) 0.9471(10) 0.1897(20) 0.024(3) Uiso 1 d . .
H1B H 0.3276(32) 0.9275(10) 0.1059(18) 0.019(3) Uiso 1 d . .
H1C H 0.2818(32) 1.0219(12) 0.2475(20) 0.027(3) Uiso 1 d . .
C2 C 0.3085(2) 0.87541(7) 0.38809(13) 0.0132(2) Uani 1 d . .
H2A H 0.1883(31) 0.8973(10) 0.5016(18) 0.017(3) Uiso 1 d . .
H2B H 0.5360(36) 0.8830(10) 0.4198(18) 0.022(3) Uiso 1 d . .
C3 C 0.2306(3) 0.75914(7) 0.34494(14) 0.0153(2) Uani 1 d . .
H3A H 0.0068(39) 0.7484(10) 0.3207(20) 0.025(3) Uiso 1 d . .
H3B H 0.3428(34) 0.7347(10) 0.2321(20) 0.018(3) Uiso 1 d . .
H3C H 0.2990(35) 0.7128(12) 0.4529(21) 0.028(4) Uiso 1 d . .
```

Figure 1: An example of CIF showing atom site coordinates for JAYDUI

A CIF file is written in a STAR Schema format that comprises data blocks identified by name-value pairs and tables as loops. The name is written as "_data_type" followed by its associated value. Figure 1 shows a part of 'JAYDUI' CIF that represents the atom's site coordinates.

The selection criteria to filter organic molecular crystals from Web CSD to create our CSD dataset are:

- Organic molecular crystals with atoms consisting of carbon, nitrogen, oxygen, phosphorous, sulfur, and halogens are selected.
- CIFs that do not have "_chemical_melting_point" block are removed.
- Molecular crystals with more than one type of molecule in the crystal and molecules in conjugate acid-base form are removed to eliminate molecules whose melting points are impacted by formal charges. Neutral-charged single-molecule crystals are only selected.

Finally, the selection criteria resulted in 63,888 molecular crystals from the CSD dataset. Melting point temperatures are converted into Kelvin.

2.2 Open Notebook Science Melting Point Dataset

The ONS Melting Points [21] curated under the supervision of Professor J.C. Bradley is an open-source dataset in the form of a comma-separated value file that represents a snapshot of melting points collected on July 30, 2011. The dataset consists of columns representing molecules in SMILES string format, name of the molecule, melting point in (°C), source of information, source's link, and csid as a unique identifier to find the molecule in the source link. Also, they have added a column "DONOTUSE" marked with "x" for molecules whose melting points are uncertain, and it is left to the user's choice to re-evaluate and update them. It contains a melting point of 28,645 molecules. The Simplified Molecular Input Line Entry System (SMILES) is a form of structure representation using ASCII string format [22]. For example, 'CCC' and 'CCO' represents a SMILES string for propane and ethanol respectively.

The selection criteria for filtering for our dataset are:

- Molecules with atoms consisting of carbon, nitrogen, oxygen, phosphorous, sulfur, and halogens are selected.
- "DONOTUSE" column whose rows are marked as "x" is removed.
- For molecules with duplicate names, we retain the first value and remove others.

The final selection has 24,889 molecules from ONS Dataset. Melting point temperatures are converted into Kelvin.

2.3 Combined Dataset

Since we aim to build a generalized model for melting point prediction, we need to evaluate the dataset in use. Figure 2 shows the histogram plot of Temperature for CSD and ONS datasets where we observe that the melting point temperature of molecules is within the range of 300 K to 600 K. Therefore if only CSD is used, our ML model will be able to learn predictions for this range of temperatures but will be unable to predict the range below 300 K as there are no molecules that represent it. The ONS has fewer molecules comparatively, but it covers the melting point temperature range of 100 K to 600 K. Therefore combining both will have a wide range of melting points from 100 K to 600 K.

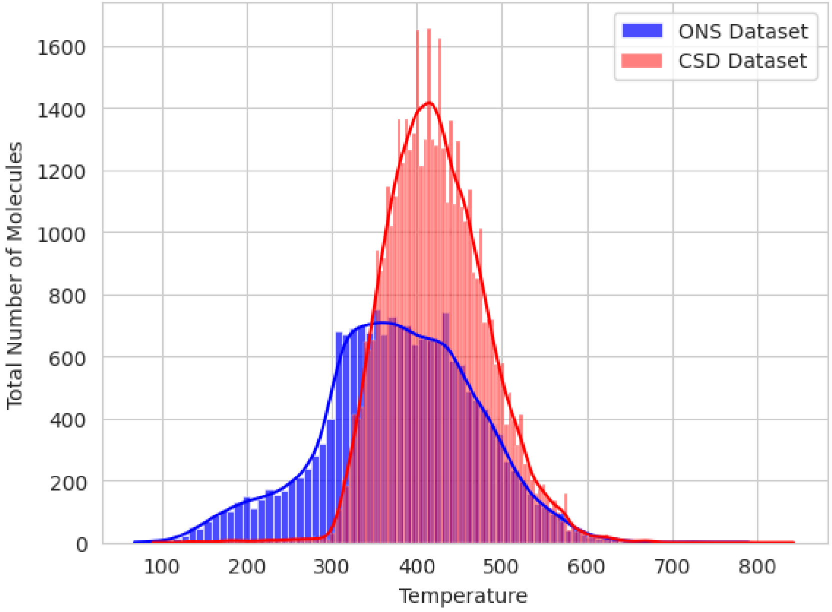


Figure 2: Histogram Plot of Temperature for CSD and ONS datasets

We can make the argument for only using the ONS dataset since it covers the temperature range of 100 K to 600 K. Therefore we evaluate both datasets based on the Bertz Connection Table(BertzCT) [23] that is a topological index describing the complexity of a molecule by accounting for size, symmetry, branching, rings, bonds, and heteroatom distribution. Figure 3 shows BertzCT's histogram plot for the CSD and ONS datasets, where the mean values are 441.16 and 758.3 for the ONS and CSD datasets, respectively. If we assume low complexity as 0-500, medium complexity as 500-1000, and high complexity as 1000+ then we observe in the figure that there are more low and medium complexity molecules in ONS than CSD that has more medium to high complexity molecules. A ML model applied to ONS will learn complex patterns and relations for low and medium-complexity molecules but will suffer from poor predictions for high-complexity molecules since its representation is less. Similarly, the model trained on CSD alone will have poor predictions for low-complexity molecules.

To further strengthen our argument for combining datasets, we extract the count of hydrocarbon molecules from both datasets represented in Figure 4. Hydrocarbons are molecules with carbon and hydrogen only. Even though there are more molecules in CSD, we can see in the figure that ONS has a higher representation of hydrocarbons. We observe that the alkanes are lower in CSD than ONS and a similar trend follows for aliphatic non-cyclic and cyclic rings and aromatic rings. It shows that ONS has a better

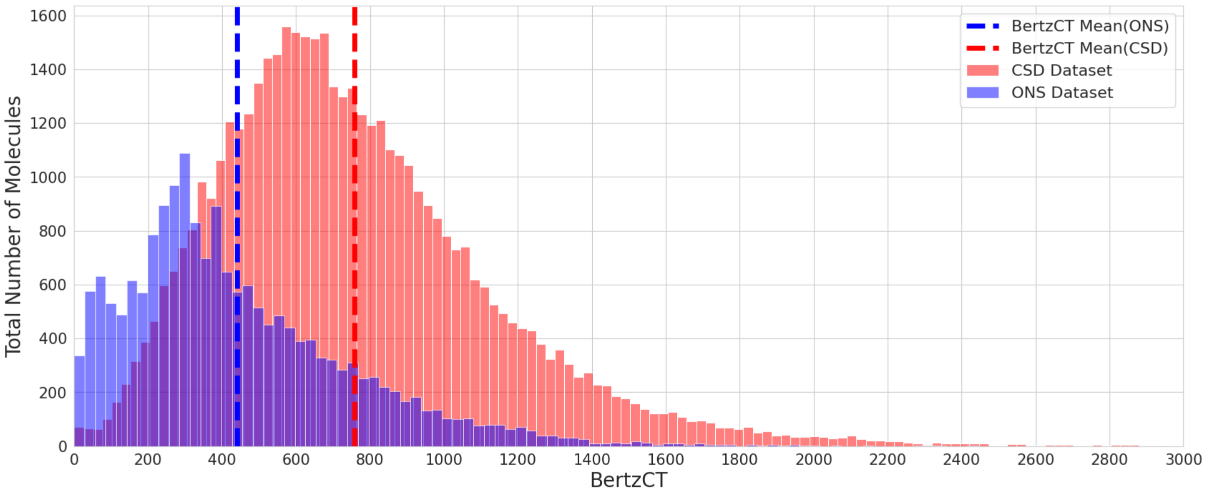


Figure 3: Histogram Plot of BertzCT for CSD and ONS Datasets

variation in molecules that if combined with CSD can form a more generalized dataset with various types of molecules.

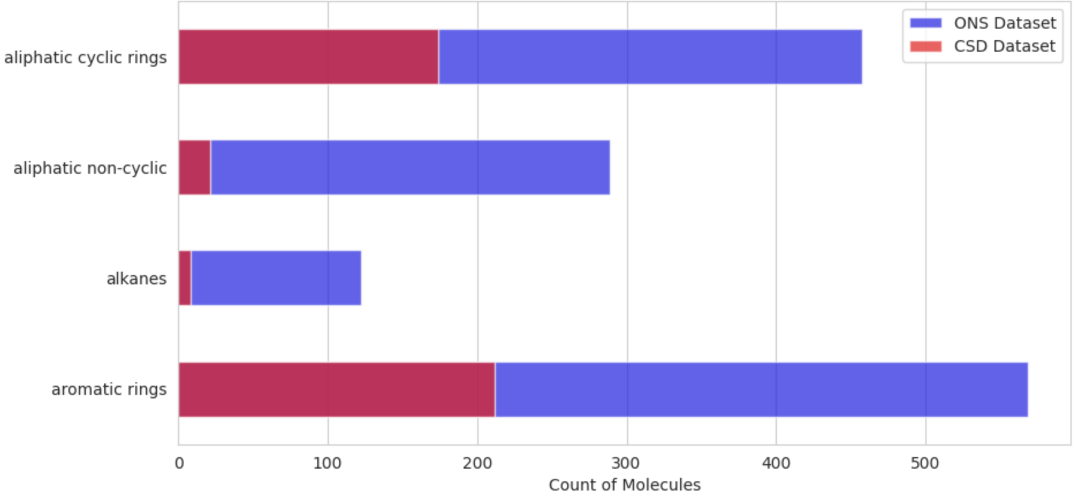


Figure 4: Bar Plot of pure hydrocarbon molecule count for CSD and ONS datasets.

Therefore combining ONS and CSD can create a more balanced dataset that represents different types of molecules, ranging from low-complexity to high-complexity. Hence, the combination now consists of 92,156 organic molecules.

We use RDKit [24] package to read the molecular structure for performing feature extraction like volume requires the computation of 3-dimensional coordinates, and SMILES strings do not have them. Even though 3D coordinates are available in CIF, RDKit cannot read CIF format but can read MOL or Structure Data File (SDF). Due to these limitations, we performed elimination in two parts - before and after combining the dataset.

Before combining the dataset, the step includes converting SMILES, or molecular crystal, to SDF format using openbabel [25] package. We use the openbabel option of separate and unique to separate disconnected fragments into individual molecular records and remove duplicate molecules. Further, the SMILES string requires to undergo generation of random 3D coordinates. SMILES or CIF that have conversion errors are removed.

Henceforward, we combine both datasets to obtain 82,298 SDF files. Further, the selection criteria to eliminate molecules from the combined dataset are as follows:

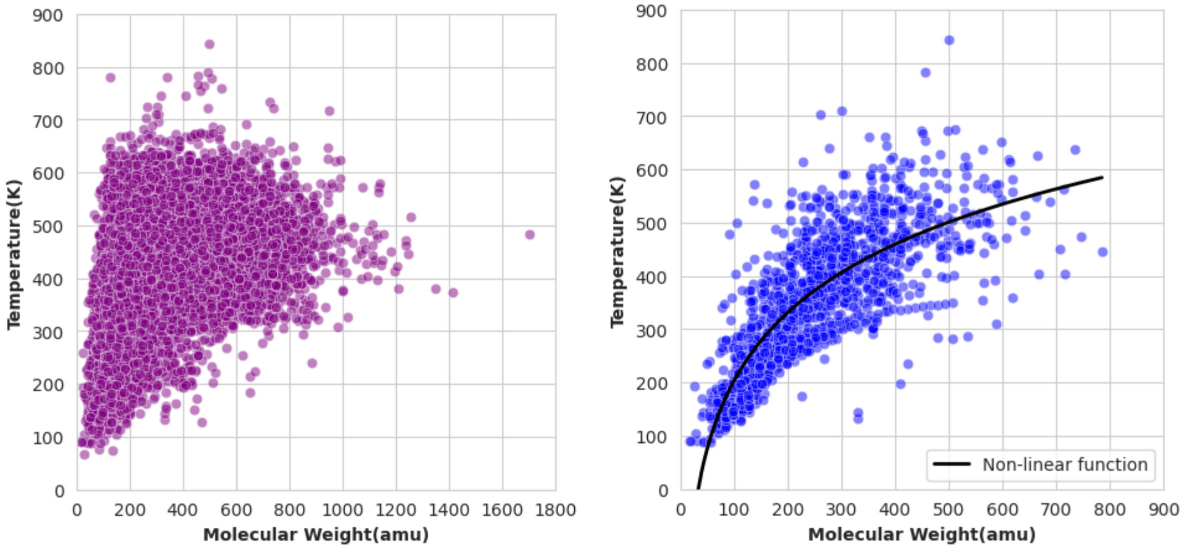
- Molecules with radical electrons greater than 0 are eliminated since they represent bond conversion errors.
- Molecules with molecular fragments of more than 1 are eliminated since we do not want multi-molecular structures.
- Molecules with a length of SDF greater than 1 are eliminated since we do not want multi-molecular structures.
- Molecules with atoms other than C, N, O, P, S, F, Cl, Br, and I are eliminated.
- Empty or corrupt SDF files are eliminated.
- Molecules with similar SMILES are eliminated as they represent duplicates.

Thus, the final combined dataset now contains 65,466 organic molecules.

2.4 Segregation of Combined Dataset

The combined dataset represents atoms forming various types of bonds, functional groups, and cyclic and aromatic rings. When we convert these properties into features on a combined dataset, we will have an obscure view of each property's contribution toward melting point temperature and be unable to make an informed decision. For example, Figure 5 shows molecular weight versus Temperature plots for the dataset and the C dataset representing hydrocarbons. In Figure 5a the combined data has information clustered for molecular weights between 0 and 1000, and we cannot determine a strong correlation between them. Whereas Figure 5b of hydrocarbons shows a strong correlation between molecular weight and temperature, and the blue line represents the function $f(x) = a \log x + b$ that shows there is a non-linear relationship. It shows that segregating the dataset into hydrocarbons allows a better view of the correlations of physical and chemical properties versus temperature.

We can further strengthen our statement by Figure 6a and 6b which show the plot of Molecular Weight (amu) versus Density(amu/Å^3) for CX (halogen compounds) and C (hydrocarbons) datasets, where density is calculated using equation (9). We observe that for halogen compounds, the density is spread across 1.0 amu/Å^3 to 3.0 amu/Å^3 whereas, for hydrocarbons, it is between 0.5 amu/ Å^3 to 1.5 amu/Å^3.



(a) Combined Dataset

(b) C Dataset

Figure 5: Plot of Molecular Weight(amu) vs Temperature(K) for Combined and C datasets.

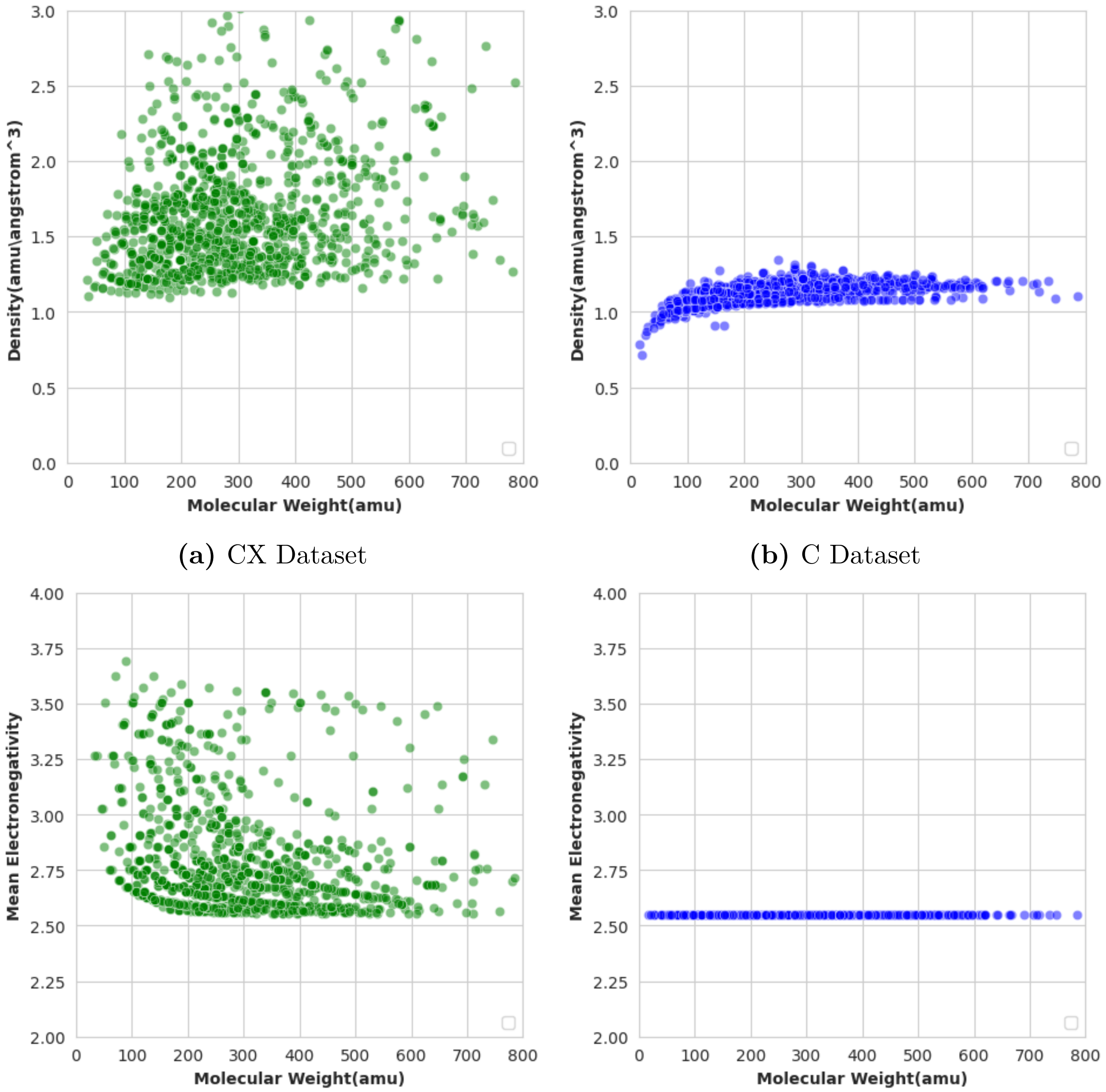
The difference in their density can be further analyzed using electronegativity. This can be viewed in Figure 6c and 6d which show the Molecular Weight(amu) versus electronegativity mean for the CX and C datasets, where electronegativity mean is calculated using equation (11). We observe that the electronegativity mean is higher for halogen compounds than hydrocarbons. Therefore, electronegativity may affect the density of the compounds. These observations would have been obscured if we had analyzed only the combined dataset. Thus, it is necessary to divide the combined dataset for a better understanding of the features and their impact on the melting points.

Therefore, we divide the dataset into four segments based on the number of bonds an atom can make.

- **C Dataset** - It contains 1,526 molecules with carbon and hydrogen atoms. Confining the focus on the "C Dataset" will assist in building features representing basic properties like molecular weight, orbital hybridization, chains, and rings.

- **CX Dataset** - It consists of 2,825 molecules having carbon, hydrogen, and halogen atoms. Halogen usually forms a single bond with carbon, and their study helps to examine one of their key properties, electronegativity. This dataset acts as a quality baseline to understand electronegativity's effect and the features we need to capture this information accurately.

- **CXOS Dataset** - This dataset has 24,265 molecules with carbon, hydrogen, halo-



(a) CX Dataset

(b) C Dataset

(c) CX Dataset

(d) C Dataset

Figure 6: Plot of Molecular Weight(amu) vs Density(amu/Å^3) and Mean Electronegativity for CX and C datasets

gen, oxygen, and sulfur. Oxygen and Sulfur atoms form two bonds, resulting in various combinations of functional groups. Thus, our goal with this dataset is to lay a foundation for studying functional groups' effects and formulate features that may even capture hydrogen bonding's effects.

- **CXOSNP Dataset** This is the combined dataset, having 65,466 molecules consisting of C, N, O, P, S, F, Cl, Br, and I. Here nitrogen and phosphorous atoms make 3 bonds, and the combinations formed with oxygen, sulfur, hydrogen, and carbon make numerous groups, and hetero rings generate highly complex molecular structures to study.

Our end goal is to attempt robust prediction for the CXOSNP dataset, and segregating as described above helps us to build features for the smaller datasets and understand its correlation with melting points.

3 Feature Engineering

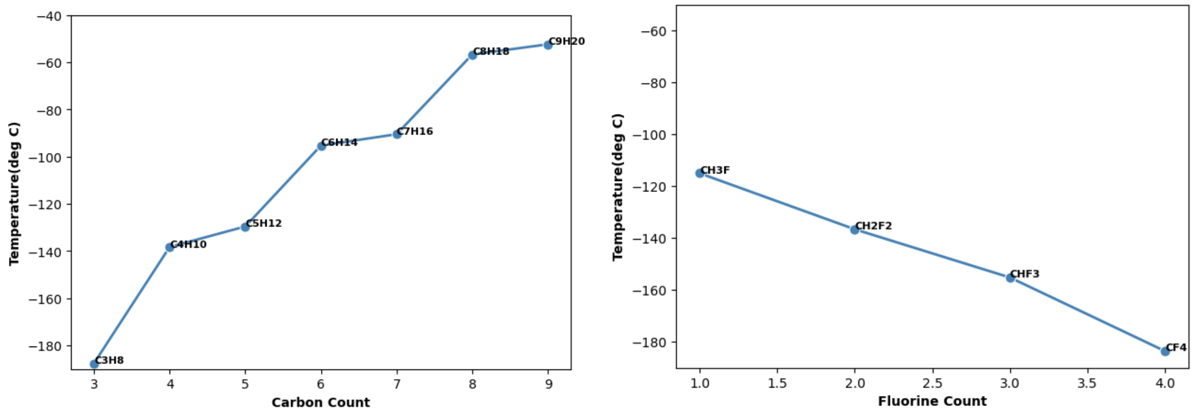
The dataset discussed in Section 2.4 consists of molecules in the form of raw data that cannot be used for training ML models. Therefore, we need to convert these molecules into numerical features using feature engineering.

Feature engineering is the process of transforming raw data into useful features by extracting relevant information from the raw data. This requires knowledge of the domain to which the dataset belongs and utilizing it to extract potential features that can assist in improving the prediction of the training model.

Thus, in this section, we will analyze the important factors that affect melting points and discuss the process of extracting them using a single feature or a combination of multiple features.

Atom Count

Figure 7a shows carbon count of alkanes: C_3H_8, C_4H_{10}, C_5H_{12}, C_6H_{14}, C_7H_{16}, C_8H_{18}, and C_9H_{20}. We observe the melting point increases with an increase in the number of carbons and has an upward trend.



(a) Alkanes

(b) Fluoroalkanes

Figure 7: Plot of Atom Count versus Temperature(°C)

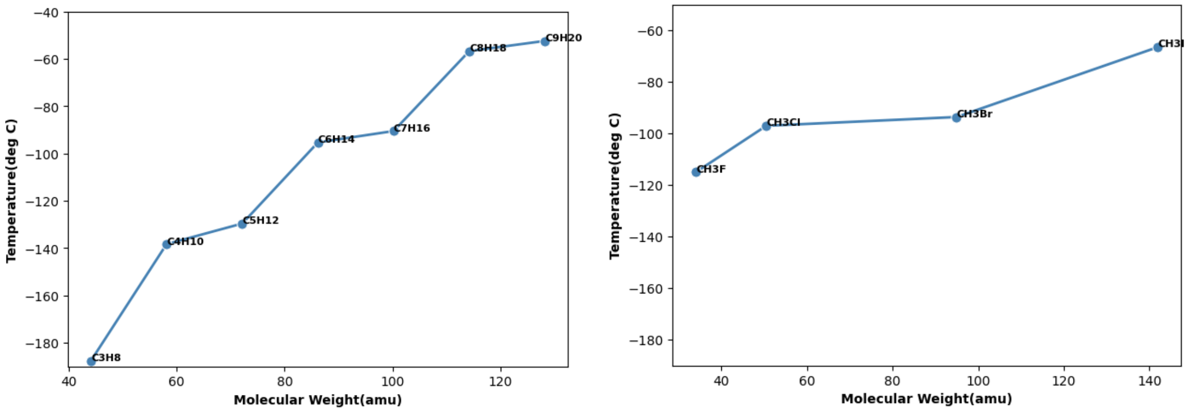
Figure 7b shows the fluorine count for fluoroalkanes: CH_3F, CH_2F_2, CHF_3 and CF_4. Here we observe that with the same number of carbon atoms and an increasing number of fluorine atoms, the melting point has a downward trend.

A simple observation on alkanes and fluoroalkanes shows that counting the number of atoms in the molecule can assist in evaluating the general trend of melting point. Even though for complex molecules the atom count alone will not be able to provide the trend, with the combination of other features, it might assist us. Therefore, we decide to create a feature to calculate the atom count for each molecule, where the atoms to be counted are C, O, S, N, P, F, Cl, Br, and I.

Molecular Weight

Molecular weight is the sum of the average atomic mass of each atom in the molecule. It is calculated using the equation (8), where m_i is the atomic mass of each atom summed together to form m which is the mass of the molecule.

$$m = \sum_{i=1}^{i=n} m_i \tag{8}$$



(a) Alkanes

(b) Haloalkanes

Figure 8: Plot of Molecular Weight(amu) versus Temperature(°C)

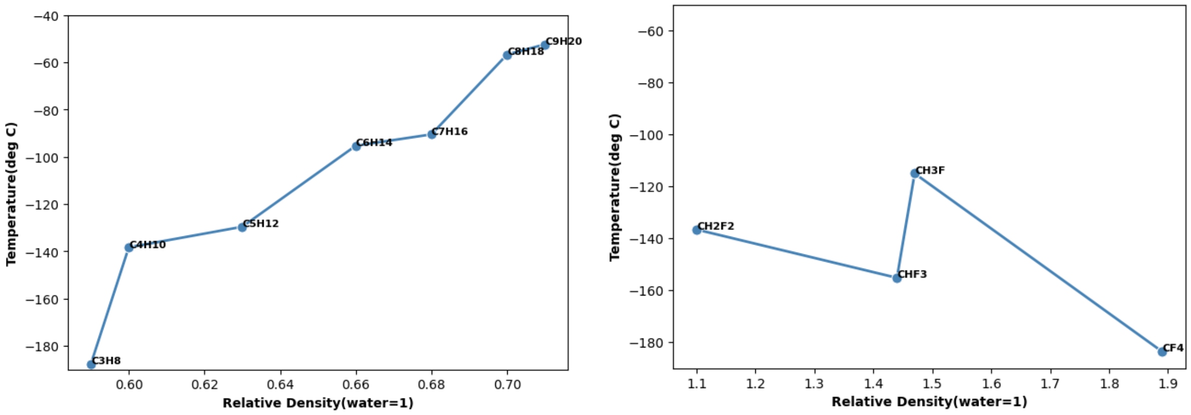
Figure 8b represents the plot of Molecular Weight(amu) versus Temperature(°C) for haloalkanes: CH_3F, CH_3Cl, CH_3Br, and CH_3I. We observe that an increase in molecular weight occurs due to the increasing weight of halogen(F $<$ Cl $<$ Br $<$ I) and therefore the melting point has an upward trend.

Figure 8a represents the plot of Molecular Weight(amu) versus Temperature(°C) for alkanes, which is similar to Figure7(a). We can argue that the count of carbons captures the same information as the molecular weight of alkanes, and for complex molecules, the count of each type of atom can collectively represent the molecular weight. The difference lies in how the model will process the information since in molecular weight(m) calculation, we pass the weight of each atom and calculate its average weight, whereas in atom count, the model has to assign the feature weight(w) to each atom, and this may depend on other features too as the model evaluates feature weight or importance based on information gain(I). Therefore, we extract both atom count and molecular weight and allow the model to assess their importance, which is evaluated during feature selection.

Density

Density is the ratio of molecular mass to the volume occupied by a crystal lattice (space-filling). It is a measure of compactness that can be calculated using the ratio of molecular weight and volume, as shown in equation (9).

$$\rho = \frac{m}{V} \tag{9}$$



(a) Alkanes **(b)** Fluoroalkanes

Figure 9: Plot of Relative Density($\rho_{\text{water}} = 1$) versus Temperature (°C).

Figure 9a shows the plot of relative density versus temperature for alkanes, where we observe that the melting point has an upward trend with an increase in density. Larger molecules tend to have a higher density, indicating better packing, which may suggest stronger intermolecular forces might exist that may require more energy to break, which increases the melting point [26].

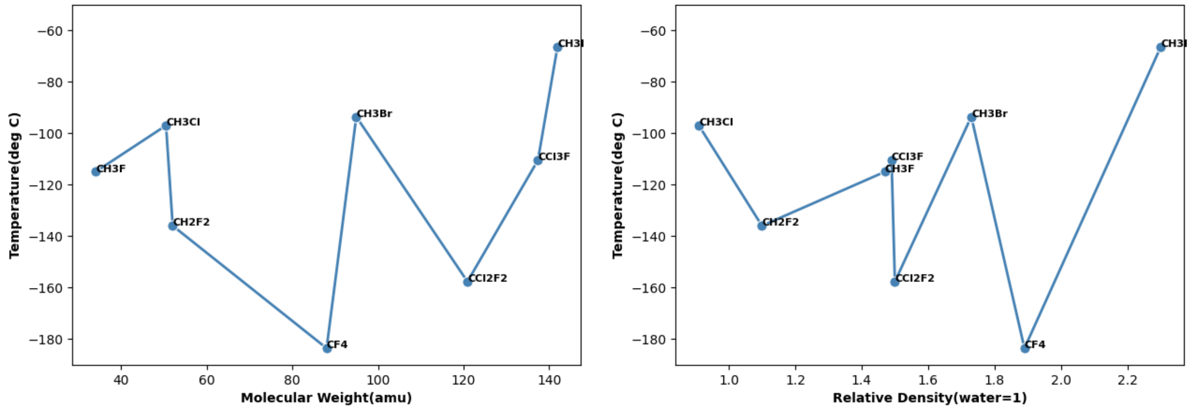
But this upward trend is not observed in Figure 9b, which represents fluoroalkanes that have an overall downward trend but a rise in the melting point from CHF_3 to CH_3F. According to the equation (9), the density of CHF_3 should have been higher since its molecular weight is greater than CH_3F, and the volume should also be higher due to 3 fluorine atoms. The CHF_3 molecule has 3 fluorine atoms that create a partial negative charge on themselves and can make one hydrogen bond, and the CH_3F molecule has only 1 fluorine atom that can make one hydrogen bond [27]. So even though they both make single hydrogen bonds, in CHF_3 there may be other factors that influence the intermolecular forces which are affecting its melting point.

Therefore, density may not always have a high correlation with the melting point but can be used as a feature that may identify the packing of atoms within the molecule.

Earlier, we defined volume as space-filling in crystal, but our dataset does not have crystal information for all molecules. Thus, we calculate vdW volume, which is the total amount of space occupied by atoms and bonds of a molecule in 3D space in its specific conformation or shape. It is calculated using Cavalieri's Principle, available in RDKit [24], which assumes a uniform electron density distribution for the molecule and creates a set of overlapping spheres based on vdW radii of the atoms and computes the total volume of the overlapping spheres. Hence the density that we calculate will be vdW density, as shown in equation (10).

$$\rho_{\text{vdW}} = \frac{m}{V_{\text{vdW}}} \tag{10}$$

Electronegativity



(a) Molecular Weight (amu) vs Temperature (°C)

(b) Relative Density($\rho_{\text{water}} = 1$) versus Temperature (°C)

Figure 10: Plot of Haloalkanes for Molecular Weight(amu) and Relative Density($\rho_{\text{water}} = 1$) versus Temperature (°C).

Electronegativity is a chemical property that allows a participating atom in a covalent bond to attract the bonding electrons, creating a partial negative charge on itself and a partial positive charge on the other participating atom, which can result in a halogen bond or hydrogen bond with its neighboring molecules.

In the periodic table, the most electronegative atoms are halogens like fluorine, chlorine, bromine, and iodine. Figure 10a shows the molecular weight of haloalkanes having 1 carbon and different combinations of halogens, where we can observe melting point does not have a linear relationship with molecular weight and density. We observe that, even though CCl_2F_2 has a larger molecular weight than CH_2F_2 the melting point is higher for CH_2F_2. For the molecules with only one halogen, the melting points are in the increasing order of $CH_3F < CH_3Cl < CH_3Br < CH_3I$ that may be due to molecular weight in the order of $F < Cl < Br < I$. But it is not in the same order in Figure 10b which shows the order of relative density as $CH_3Cl < CH_3F < CH_3Br < CH_3I$.

These observations indicate that electronegativity induced in the molecule can be partially captured for a few molecules using molecular weight and density trends, but not for all complex molecules. Since we want to calculate the polarity induced in the atoms of a molecule and represent them as a singular numerical feature to capture information about halogen and hydrogen bonds, we use the below methods to identify electronegativity trends.

- Equation (11) calculates the mean of electronegativity of all atoms within the molecule, where χ_i is the electronegativity of an atom i and N is the total number of atoms. If there are more electronegative atoms in the molecule, the mean will be higher than in a molecule with only carbon atoms.

$$\text{EN\_mean} = \frac{1}{N}\sum_{i=0}^{N}\chi_i \tag{11}$$

- Equation (12) calculates the variance of electronegativity of a molecule, where χ_i is the electronegativity of an atom i, μ is the mean, and N is the total number of atoms. This will be able to represent the deviations in electronegativity when there is a higher electronegative atom but its count is low.

$$\text{EN\_var} = \frac{1}{N}\sum_{i=0}^{N}(\chi_i - \mu) \tag{12}$$

- Equation (13) calculates the mean of the electronegativity difference between two adjacent atoms in a molecule. χ_i and $\chi_i^{'}$ represent electronegativity for atoms for connection i, and n is the total number of connections between atoms in the molecule. Here, the electronegativity difference will be 0 if the adjacent atoms are C-C, but the difference will rise if they are C-F. The mean of these adjacent atoms can capture the polarity that may be induced between the atoms.

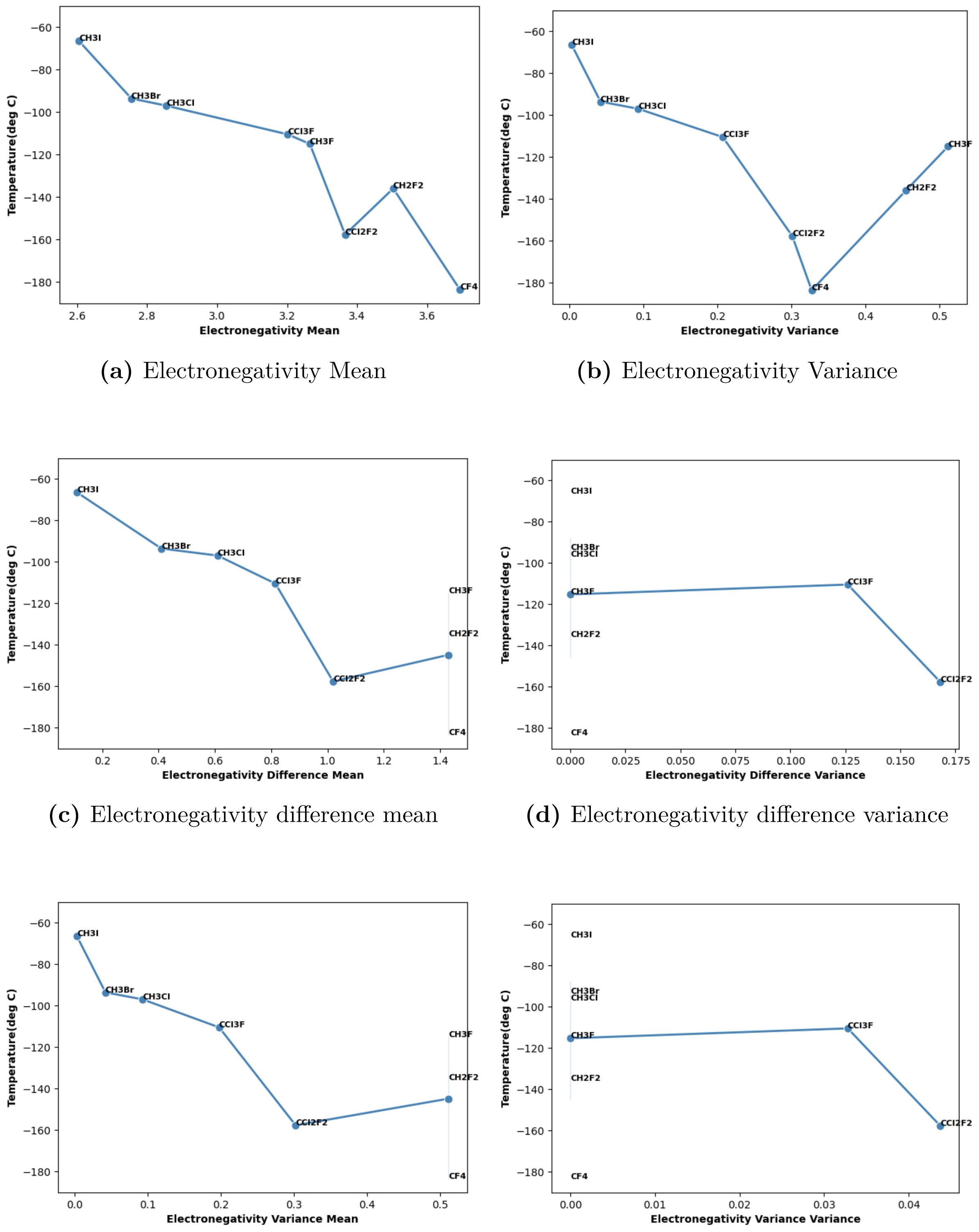
$$\text{EN\_diff\_mean} = \frac{1}{n}\sum_{i=0}^{n}(\chi_i - \chi_i^{'}) \tag{13}$$

- Equation (14) calculates the mean of electronegativity variance between χ_i and $\chi_i^{'}$ for connection i in a molecule. Its purpose is similar to equation (13) but we try to capture the information between adjacent atoms using variance.

$$\text{EN\_var\_mean} = \frac{1}{n}\sum_{i=0}^{n}\text{Var}(\chi_i, \chi_i^{'}) \tag{14}$$

- Equation (15) calculates the variance of electronegativity variance between χ_i and $\chi_i^{'}$ for connection i in a molecule. Its purpose is similar to equation (14) but captures information using a variance.

$$\text{EN\_var\_var} = \text{Var}\left(\text{Var}(\chi_i, \chi_i^{'})\right) \tag{15}$$



(a) Electronegativity Mean

(b) Electronegativity Variance

(c) Electronegativity difference mean

(d) Electronegativity difference variance

(e) Electronegativity variance mean

(f) Electronegativity variance variance

Figure 11: Plot of different electronegativity calculations versus Temperature (°C)

- Equation (16) calculates the variance of electronegativity difference between between χ_i and $\chi_i^{'}$ for connection i in a molecule. It is a combination of equation (13) and (15).

$$\text{EN\_diff\_var} = \text{Var}(\chi_i - \chi_i^{'}) \tag{16}$$

Figure 11 shows the plot of different electronegativity calculations versus Temperature (°C). In Figure 11a and 11b, we observe a downward trend that can capture the electronegativity in the molecule. For Figure 11d and 11f, there is no trend and the results may not assist ML model to gain a significant amount of information from this feature.

Thus, these electronegativity functions will capture polarity-induced, and assist to identify the presence of intermolecular bonding between the molecules.

Orbital Hybridization

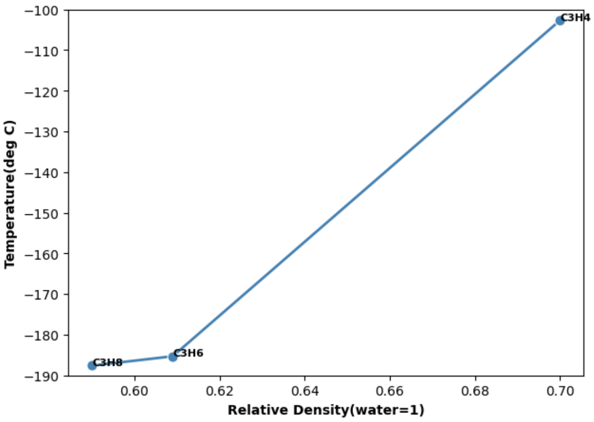


Figure 12: Plot of Relative Density ($\rho_{\text{water}} = 1$) versus Temperature (°C) for Propane (C_3H_8), Propylene (C_3H_6) and Propyne (C_3H_4).

Hybridization is the process of redistribution of the orbital energy of individual atoms to give rise to hybrid orbitals that have different energy, shapes, etc. for the pairing electrons to form chemical bonds. In simple hydrocarbons, we observe three types of hybridization: sp, sp2, and sp3. sp3 hybridization occurs when one electron from the s orbital and three electrons from the p orbital mix together to form four hybrid orbitals in a tetrahedron shape, making an angle of 109°28′ with one another. In sp2 hybridization, one s orbital and two p orbitals join together to form three hybrid orbitals, forming a triangular planar shape with an angle of 120°. In sp hybridization, the valence shell contains one unpaired electron in the s and p orbitals, which participate in forming the sp orbitals. This mixing of s and p orbitals produces two identical sp orbitals that form a linear shape with an angle of 180°.

Thus, the main effect on melting that comes from orbital hybridization is the shape factor of the molecule. Since hybridization determines the shape of the molecule, it thus contributes to the molecule packing arrangement in the crystal lattice and the intermolecular forces that occur due to the arrangement. This is observed in Figure 12 which shows the relative density versus melting point of C_3H_8, C_3H_6, and C_3H_4. We can observe that the density has an upward trend where C_3H_4 being a linear shape, has the highest density due to better packing leading to stronger intermolecular forces, thus having a higher melting point than others. Since C_3H_8 has a tetragonal shape, the packing may not be sufficient for stronger intermolecular forces, leading to a lower melting point.

Hence, from these observations, we understand that hybridization impacts the shape of the molecule, which may influence many factors like polarity, rotation, expansion, flexibility, etc. that impact the melting point.

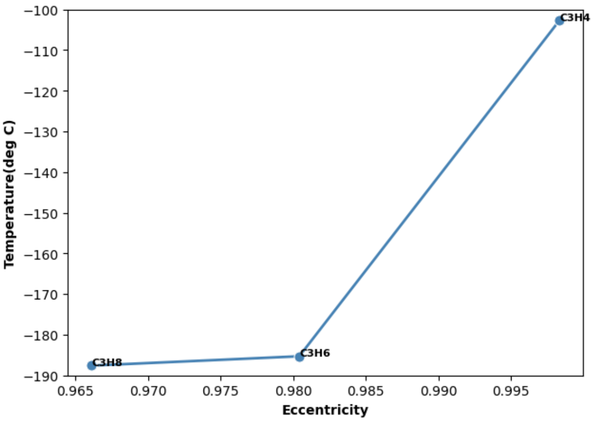


Figure 13: Plot of Eccentricity vs versus Temperature(°C) for Propane(C_3H_8), Propylene(C_3H_6) and Propyne(C_3H_4)

For complex molecules, density alone will not be able to capture the information of hybridization. Therefore, we chose the below features to capture the hybridization and the shapes that may arise due to it.

- We calculate the total number of bonds in a molecule and the individual types of bonds that are single, double, triple, and aromatic. These features will be able to describe the type of connections between the atoms, and collectively, they can describe the hybridization.
- We attempt to capture flexibility using rotatable bonds that are single bonds not part of a cyclic structure, and occur between two carbon or heteroatoms separated by three or more bonds. The last condition is required to ensure the bond is relatively flexible and not constrained by small rings or other structures around it.
- To capture flatness in the molecule, we use eccentricity. Eccentricity describes the flatness of a geometric shape given by a number between 0 and 1, where 0 corresponds to a circle and 1 is a line segment. According to Tosdeschini and

Consoni's "Descriptors from Molecular Geometry" [28] eccentricity can be estimated by the equation (17) where pm_1 and pm_3 represent the principal moments of inertia 1 and 3, respectively, where moments are calculated with atomic weight.

$$E = \frac{\sqrt{(\text{pm}_3)^2 - (\text{pm}_1)^2)}}{\text{pm}_3} \tag{17}$$

Figure 13 shows the plot of Eccentricity versus Temperature(K) for the same molecules, where we can capture the flatness of C_3H_4 as it has the highest eccentricity.

Chains

Straight chains are hydrocarbons that are connected in a continuous linear chain and Branched chains are hydrocarbons that have groups branching out from the connected straight chains. Fig 14 represents isomers of hexane, where n-Hexane represents a straight chain and other isomers represent a branched chain.

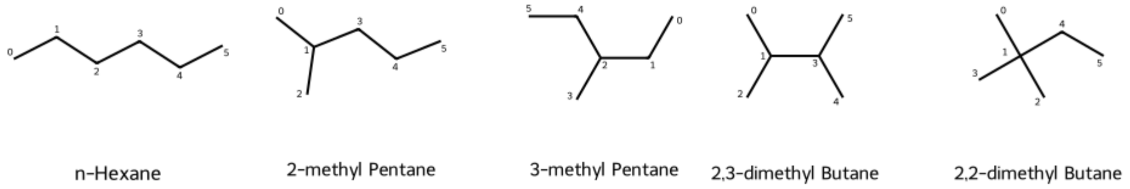
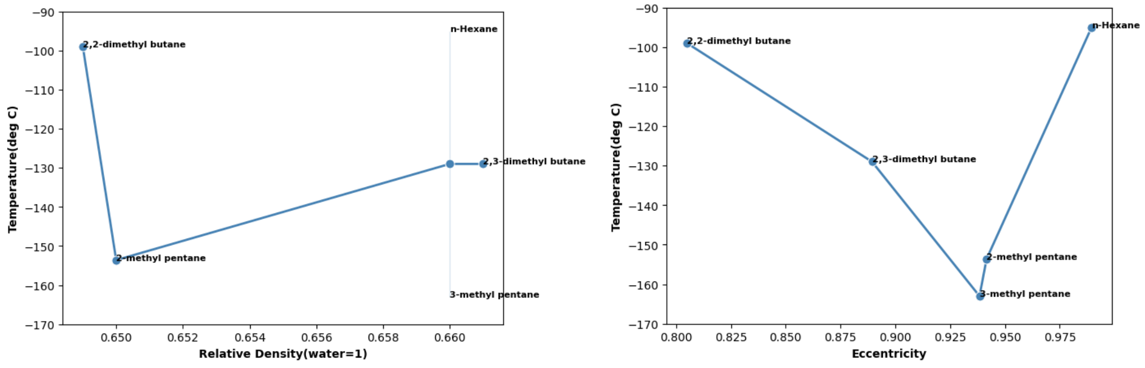


Figure 14: Molecular Graphs of Isomers of Hexane



(a) Relative Density vs Temperature

(b) Eccentricity vs Temperature

Figure 15: Plot of Relative Density and Eccentricity versus Temperature for Isomers of Hexane.

Figure 15a shows the plot of relative density versus temperature for isomers of hexane, where we observe that 2,2-dimethyl butane has the lowest relative density and 2,3-dimethyl butane has a higher density. Also, we can observe that n-hexane and 3-methyl pentane have the same density at different melting points. Thus, density is not a strong indicator of melting point trends for hexane isomers.

Figure 15b shows the plot of eccentricity versus temperature where we observe melting point decreases with increasing eccentricity from 2,2-dimethyl butane to 3-methyl pentane. From 3-methyl pentane to n-hexane, the melting point increases with eccentricity. Since n-hexane is a straight-chain molecule and has the highest eccentricity, which means it is a more flat molecule and will have better packing compared to its branched isomers and therefore has the highest melting point.

2,2-dimethyl butane's melting point is closer to n-hexane, which cannot be explained by eccentricity. Branching creates irregular molecules that may not pack better, but when the branching leads to a more sphere-like structure, the molecules can again pack better, which can increase intermolecular forces. Therefore, we need to identify the shape of 2,2-dimethyl butane as spherical to explain its higher melting point compared to other isomers.

We calculate the sphericity of a molecule according to Tosdeschini and Consoni's "Descriptors from Molecular Geometry" [28], which describes how spherical or non-spherical a geometric shape is. It represents a number between 0 and 1, where 0 corresponds to a highly non-spherical molecule and 1 represents a highly spherical molecule. It is given by the equation (18) where pm_1, pm_2, and pm_3 are the principal moments of inertia 1, 2, and 3, respectively, and moments are calculated without atomic weight.

$$S = 3 \times \frac{\mathrm{pm}_1}{\mathrm{pm}_1 + \mathrm{pm}_2 + \mathrm{pm}_3} \tag{18}$$

Figure 16 shows the plot of sphericity vs temperature for isomers of n-hexane, where we observe that 2,2-dimethyl butane is the most spherical molecule among the isomers. Sphericity also shows that n-hexane is the least spherical molecule, which relates to Figure 15b which shows it is the most flat molecule.

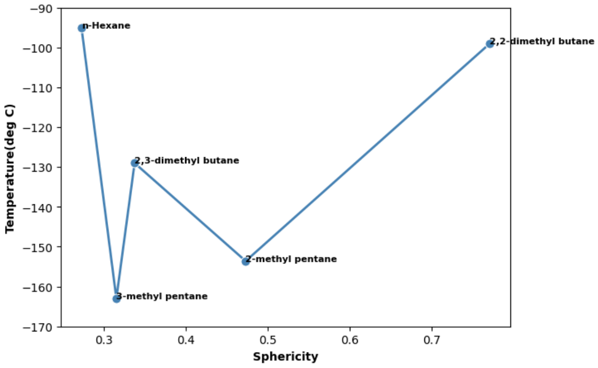


Figure 16: Plot of Sphericity vs Temperature for Isomers of Hexane.

Therefore, both eccentricity and sphericity combined can capture molecular information that represents the shape as linear and spherical, respectively.

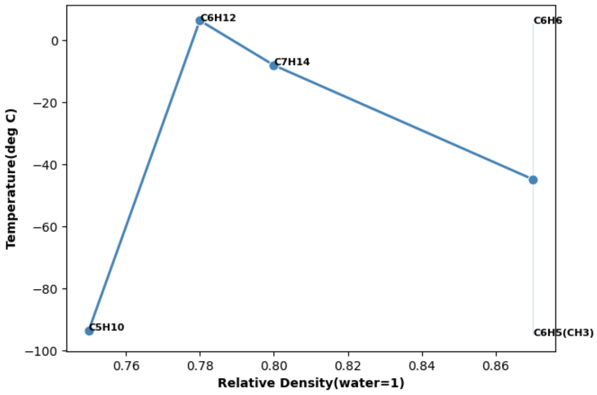


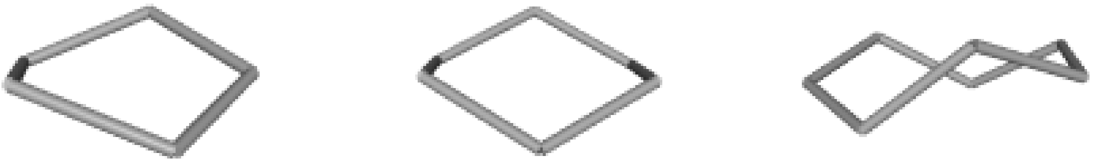
Figure 17: Plot of Relative Density($\rho_{\text{water}} = 1$) vs Temperature(K) for C_3H_6 (cyclopropane), C_4H_8 (cyclobutane), C_5H_{10} (cyclopentane), C_6H_{12} (cyclohexane), C_7H_{14} (cycloheptane), C_6H_6 (benzene), and $C_6H_5CH_3$ (toluene).

Rings

A ring comprises three or more atoms connected in a loop to form a ring structure. Figure 17 shows the plot of relative density versus melting points of cycloalkanes and aromatic compounds.

Cycloalkanes comprise carbon and hydrogen only, with carbons forming a single bond with each other, thus creating a cyclic ring. In Figure 17, we observe that the melting point does not have a linear relationship with density. We also observe that C_5H_{10} has the lowest melting point and density, and C_6H_{12} has the highest melting point even though its density is less than C_7H_{14}.

This may occur due to ring strain, which creates a more deformed shape for C_7H_{14} as seen in Figure 18 compared to C_6H_{12} that has a chair conformation shape, which leads to better packing. The shape of a cyclic ring is determined by the ring strain caused by the deformation of bond angles and lengths from their ideal values, leading to an increase in the energy of the molecule. Therefore, to reduce this strain, the molecule forms an irregular shape that reduces the energy due to the constraints. Thus, in cyclic rings, the ring strain affects the shape, which may influence the melting point.



(a) Cyclopentane (b) Cyclohexane (c) Cycloheptane

Figure 18: Wire Frame structure of CycloAlkanes [1]

An aromatic ring is a cyclic structure having resonances of alternating single and double bonds with delocalized pi electrons above and below the plane of the ring, and the shape is planar. We can assume the planar structure of benzene will have a higher melting point than the chair conformation structure of cyclohexane based on shape, but in Figure 17, we observe cyclohexane has a higher melting point of 6.4°C than benzene, with a melting point of 5.5°C. Hence, shape alone does not influence the melting points since, in this case, there are vdWs forces and electrostatic contributions that lead to higher melting points.

But between benzene and toluene, benzene has a higher melting point due to the extra methyl group in toluene, leading to weaker packing and decreasing the intermolecular forces [26].

Heteroaromatic compounds have at least one heteroatom in the aromatic ring. Consider C_4H_4O (furan), C_4H_4S (thiofuran), and C_4H_4NH (pyrrole) whose melting points are -85.61 °C, -39.4 °C, and -24 °C, respectively that have the same number of carbons and one heteroatom in the aromatic ring. According to Pauline's electronegativity, oxygen is 3.44, nitrogen is 3.04, and sulfur is 2.58 then the heteroaromatic compound's melting point should be in the order $C_4H_4O < C_4H_4NH < C_4H_4S$ but actual observed values of the melting points are in the order of $C_4H_4O < C_4H_4S < C_4H_4NH$.

In C_4H_4NH, nitrogen induces polarity with its neighboring hydrogen, creating a partial positive charge that may form hydrogen bonding, which is a stronger intermolecular bond compared to other molecules that may only induce vdW forces that might have resulted in a higher melting point than others.

Hence from these observations, shape, aromaticity, electronegativity, and hydrogen bonding may influence the melting point. We can capture shape using eccentricity and types of bonds, but we need to even capture the distinction between the types of rings in the molecule.

We capture the type of ring by representing the feature as the ring count of a molecule. The type of ring can be determined by the number of atoms that form it, like a 3-ring, 5-ring, or 10-ring. Although this representation can capture the size, it does not provide detail on whether it is an aromatic or cyclic ring. If we include the type of ring with a count like 3-cyclic-ring or 5-aromatic-ring, the dataset will consist of features with many values of zero since not all types of rings will be present in each molecule, thus creating a sparse dataset. The problem with a sparse dataset is that the information gained in each column is less and requires a larger combination of columns to represent the same. Therefore, we chose the simple approach of categorizing rings into four types, as mentioned below:

- Aliphatic Carbo Rings - Cyclic rings that only comprise carbons and at least one non-aromatic bond.
- Aliphatic Hetero Rings - Cyclic rings that comprise carbon and at least one heteroatom. Also, it must contain at least one non-aromatic bond.
- Aromatic Carbo Rings - Aromatic rings that only comprise carbons, and all bonds

in the rings must be aromatic bonds.

- Aromatic Hetero Rings - Aromatic rings that comprise carbon and at least one heteroatom. All bonds must be aromatic bonds.

Name	Molecular Formula	Aliphatic Ring Count	Single Bond Count	Double Bound Count	Carbon Count
cyclobutane	C_4H_8	1	4	0	4
cyclobutene	C_4H_6	1	3	1	4
cyclopentane	C_5H_{10}	1	5	0	5

Table 1: Example of Ring Count for cyclobutane, cyclobutene, and cyclopentane. A combination of features can distinguish between molecules with similar ring counts.

We also count the total number of rings present in the molecule as a sum of all four types. Even though this does not contain the number of atoms that form the ring, the information is preserved in the combination of bond, atom, and ring count, as shown in Table 1.

Hydrogen Bonding

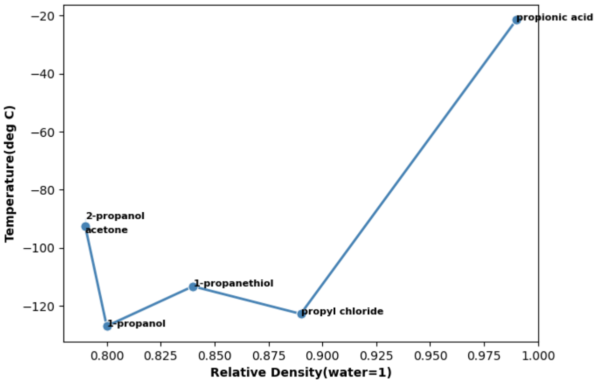


Figure 19: Plot of organic compounds having three carbons and different functional groups in ascending order.

When an electronegative atom is covalently bonded to a hydrogen atom there is a difference in electronegativity that creates a partial positive charge on the hydrogen atom and a partial negative charge on the electronegative atom and allows the hydrogen atom to form an intermolecular hydrogen bond with an electronegative atom of another molecule. Hydrogen bonding is weaker than covalent bonds but stronger than vdW forces.

The strength of hydrogen bonding is visible in compounds with functional groups like carboxyl, hydroxyl, ketone, etc where the general order of melting point according to the strength of hydrogen bonding is $CONH_2 > COOH > OH > NH_2 > NO_2 > I > Br > Cl > F > CH_3$ [14]. It can be observed in Figure 19, which shows the melting points of organic compounds having 3 carbons with different functional groups. Propionic acid having a carboxyl(COOH) group has the highest melting point of -21.5°C which is consistent with the general order but 1-Propanol having hydroxyl(OH) group has a lower melting point of -127°C than Propyl Chloride having -122.8°C which is not consistent. Also, we observe the isomer of 1-Propanol is 2-Propanol and has a higher melting point of 90°C that may arise from oxygen having a central position in the molecule allowing better hydrogen bonding. So hydrogen bonding strength may depend on the type of functional group but the melting points can vary based on its position in the molecule also.

Fragments represent a substructure present in the molecule. As discussed in Section 3, identifying such substructure allows a better representation of functional groups involved in hydrogen bonding. We use the RDKit's Fragment Module [29] and attempt to identify all types of fragments available as part of the module. The fragments are named "N_{fragment}" where fragment name can be like "Al_OH" which is an aliphatic hydroxyl group, or "Ar_OH" which is an aromatic hydroxyl group. Thus we create 64 features representing all types of fragments.

Balaban J Index

Balaban J Index is a topological index suggested by Alexandru.T Balaban in "Topological Indices based on Topological Distances in Molecular Graphs" [30]. The Balaban J index in RDKit [31] is calculated using equation (19) [32], where the molecule is represented as connected graph G having m and n as the degrees of vertex and edge set of G, respectively. $w(u)$ and $w(v)$ represent the sum of distances from vertex u and v to other vertices of G.

$$J_{\text{mol}}(G) = \frac{m}{m-n+2} \sum \frac{1}{\sqrt{w(u) \cdot w(v)}} \tag{19}$$

We include this graph descriptor since it describes the shape of the molecule using graph theory and acts as a feature to identify the complexity of the molecule's shape.

Table 2 shows the list of features created.

Feature Type	**Feature Name**
Atoms	C count N count O count P count S count F count Cl count Br count I count
Molecular Weight (amu)	Molecular Weight
vdW Volume (Å^3)	Volume
vdW Density (amu/Å^3)	Density
Bonds	Single Bonds Double Bonds Triple Bonds Aromatic Bonds Total Bonds
Electronegativity	Electronegativity mean (EN_mean) Electronegativity variance (EN_var) Electronegativity difference mean (EN_diff_mean) Electronegativity difference variance (EN_diff_var) Electronegativity variance mean (EN_var_mean) Electronegativity variance variance (EN_var_var)
Rings	Aliphatic Carbo Rings Aromatic Carbo Rings Aliphatic Hetero Rings Aromatic Hetero Rings Total Rings
Shape	Eccentricity Sphericity Balaban J Index
Flexibility	Rotatable Bonds
Functional Groups and Fragments (sub-structures) to detect Intermolecular forces	fr_Al_COO fr_Al_OH fr_Ar_COO fr_Ar_OH fr_aldehyde fr_benzene fr_Ar_N fr_Ar_NH fr_Imine fr_NH0 fr_NH1

Feature Type	Feature Name
	fr_NH2
	fr_N_O
	fr_Ndealkylation1
	fr_Ndealkylation2
	fr_HOCCN
	fr_Nhpyrrole
	fr_C_S
	fr_SH
	fr_alkyl_carbamate
	fr_alkyl_halide
	fr_aryl_methyl
	fr_amide
	fr_amidine
	fr_aniline
	fr_azide
	fr_azo
	fr_barbitur
	fr_bicyclic
	fr_epoxide
	fr_ester
	fr_ether
	fr_furan
	fr_guanido
	fr_hdrzone
	fr_hdrzine
	fr_imidazole
	fr_isocyan
	fr_ketone
	fr_lactone
	fr_methoxy
	fr_morpholine
	fr_nitrile
	fr_nitro
	fr_nitro_arom
	fr_nitroso
	fr_oxazole
	fr_oxime
	fr_phenol
	fr_phos_acid
	fr_phos_ester
	fr_piperdine
	fr_piperzine
	fr_priamide
	fr_pyridine

Feature Type	Feature Name
	fr_sulfide fr_sulfonamd fr_sulfone fr_term_acetylene fr_tetrazole fr_thiazole fr_thiocyan fr_thiophene fr_urea

Table 2: List of 96 Features created after Feature Engineering.

Embedding Features

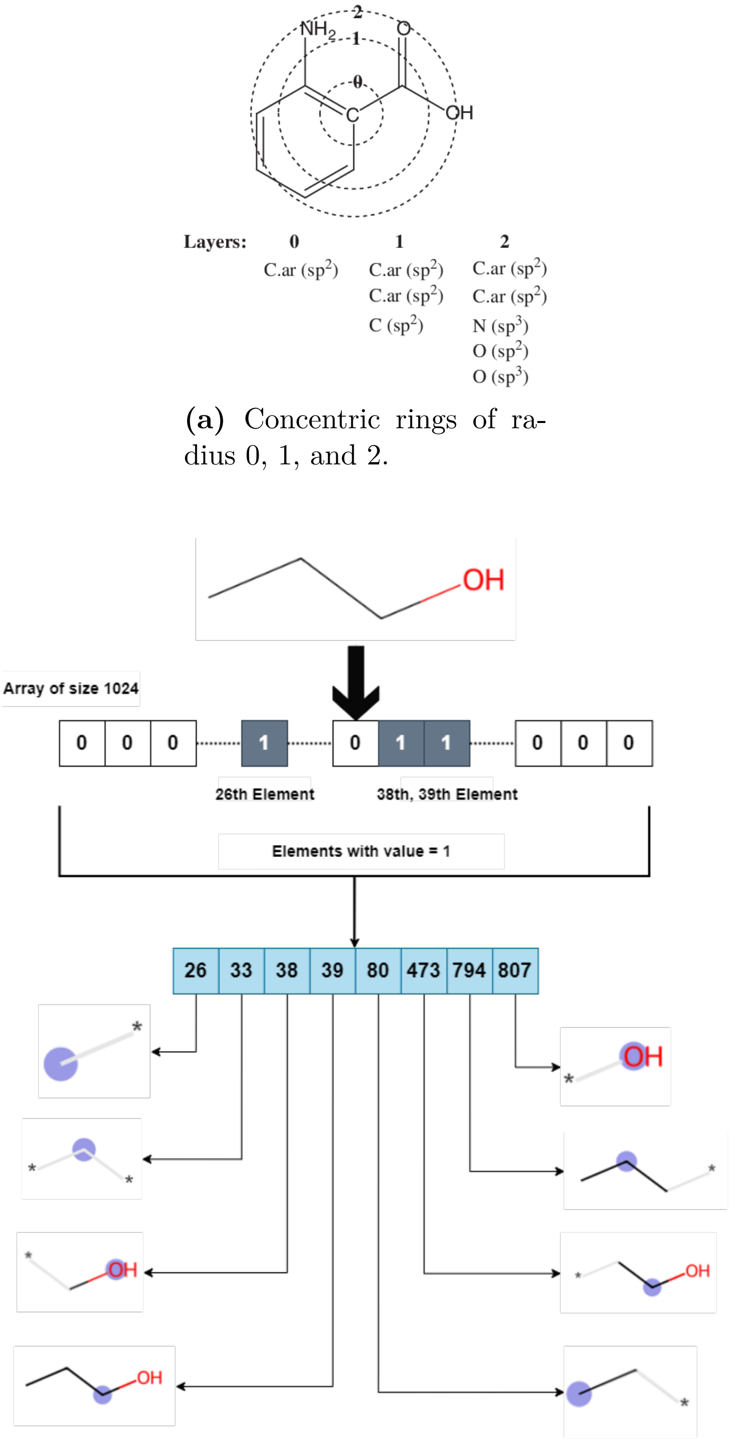
Embedding features represent molecules as an array, with each element of the array representing a substructure pattern existing in the molecule. If the substructure pattern exists, then the element value is 1; otherwise, the value is 0. An example of a substructure pattern is O-H, C-C, or C=C-C, where if the molecule has any of these structures present, then the array element that represents the specific structure will have a value of 1.

We will discuss this in detail for the two embedding features: Morgan Fingerprint and Mol2vec, which we will use on our dataset and test if our feature can outperform them.

Morgan Fingerprints [33] is an extended circular fingerprint using a variant of the Morgan algorithm [34], to represent the structural information of a molecule as a binary string in an array. It is widely used in chemoinformatics applications such as similarity search [35], drug discovery [36], toxicity [37], and classification [38].

The algorithm initially assigns each atom an identifier, which can be its atomic number, that forms the fingerprint set. Then the molecule is divided into a series of concentric rings of a desired radius centered on each atom, as shown in Figure 20a. Each atom collects its identifier and its neighbor's identifier into an array, which is converted into a new single-digit identifier using a hash function that represents the fragment. Once all atoms generate their new identifiers, the old fingerprint set is replaced with these new identifiers. This iteration is repeated a prescribed number of times to remove duplicate identifiers in the fingerprint set. The final output is the remaining identifiers, representing unique fragments present in the molecule.

We can understand Morgan fingerprint conversion through Figure 20b, where we convert 1-propanol into a 1024-bit array with a radius of 1 using RDKit [24], resulting in a vector of 0s and 1s. The fingerprint set consists of 8 substructures filled with a value of 1 on the element numbers: 26, 33, 38, 39, 80, 473, 794, and 807. Each substructure uniquely identifies every possible pattern in the molecule. If we change the array size to 256, then the element numbers will also change but represent the same set of 8 substructures.

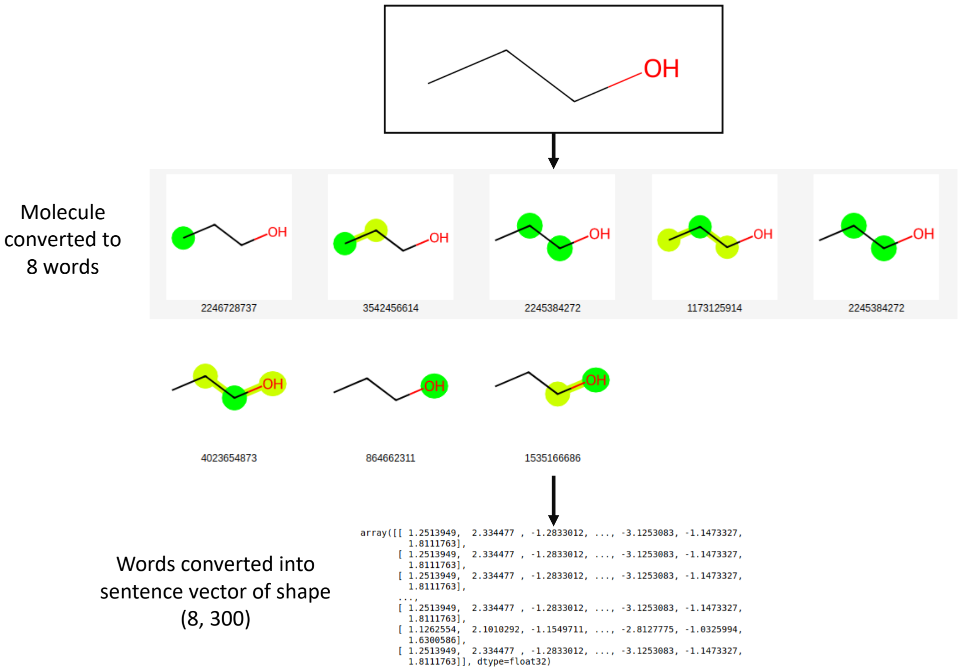


(a) Concentric rings of radius 0, 1, and 2.

(b) Conversion of Propanol to Morgan fingerprint array

Figure 20: An example of Morgan Fingerprint. (a) shows an illustration of concentric rings of radius 0, 1, and 2 formed by the Morgan fingerprint algorithm to identify substructures within the given radius [2]. (b) shows an illustration of the conversion of 1-Propanol to Morgan Fingerprint of a 1024-bit array with 8 unique substructures identified.

Mol2vec is an unsupervised ML approach to representing molecule substructures into a vector representation that is inspired by the Natural Language Processing (NLP) of Word2Vec. Mol2Vec has been used in regression [39] and classificationc [39] tasks for solubility and toxicity. The model is pre-trained on ZINC [40] and ChEMBL [41] databases with atom counts between 3 and 50, which include the atoms H, B, C, N, O, F, P, S, Cl, and Br, to identify all possible molecular substructures of radii 0 and 1. Analogous to NLP, molecules are represented as sentences, and substructures are represented as words.


OH
Molecule converted to 8 words
2246728737
3542456614
2245384272
1173125914
2245384272
4023654873
864662311
1535166686
Words converted into sentence vector of shape (8, 300)

```
array([[ 1.2513949,  2.334477 , -1.2833012, ..., -3.1253083, -1.1473327,
         1.8111763],
       [ 1.2513949,  2.334477 , -1.2833012, ..., -3.1253083, -1.1473327,
         1.8111763],
       [ 1.2513949,  2.334477 , -1.2833012, ..., -3.1253083, -1.1473327,
         1.8111763],
       ...,
       [ 1.2513949,  2.334477 , -1.2833012, ..., -3.1253083, -1.1473327,
         1.8111763],
       [ 1.1262554,  2.1010292, -1.1549711, ..., -2.8127775, -1.0325994,
         1.6300586],
       [ 1.2513949,  2.334477 , -1.2833012, ..., -3.1253083, -1.1473327,
         1.8111763]], dtype=float32)
```

Figure 21: Illustration showing conversion of 1-Propanol to Vector shape (8, 300) using Mol2vec

Figure 21 shows an illustration of the conversion of propanol using the Mol2Vec algorithm. Mol2Vec uses the Morgan fingerprint to identify the substructures present in the molecule and compares them to the substructures present in its pre-trained model. The model represents each substructure with a unique numerical number. This is similar to a bag of words in NLP. Once all features are provided with their numbers, the algorithm converts the molecule into the shape of an (8, 300) array, which is a hashing method to represent the substructure as a molecular sentence. This can then be provided to an ML model for regression or classification tasks.

Since we are not making modifications to these features, which are proven to perform well on melting point prediction for organic compounds [42], we will use their performance as benchmarks and compare their results with our descriptive features.

4 Machine Learning Theory

ML is a branch of Artificial Intelligence (AI) that involves training algorithms to learn from known data and utilizing the knowledge gained by the algorithm to make predictions on unknown data. These algorithms involve statistical methods to learn from input features and samples to identify patterns and make decisions about the target features and samples based on the learned patterns. It is widely used in classification and regression tasks.

ML is divided into three major categories - supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the ML model is provided both input features and target features to predict. This enables the ML to learn by creating a function from the input variables and prediction is performed by utilizing the learned function. In unsupervised learning the ML model is provided the input features only and the goal is to find patterns in the input features. In reinforcement learning, the ML model learns by a feedback mechanism when it interacts with the data where it is penalized for wrong actions and rewarded for correct actions.

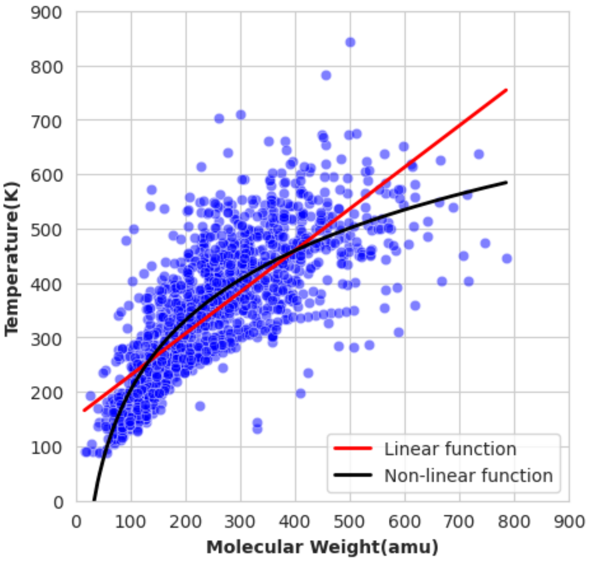


Figure 22: Plot of Molecular Weight(amu) vs Temperature(K) with regression lines using linear($y = ax + b$) and non-linear($y = a\log(bx) + c$) functions. The non-linear function is a more approximate fit to the data points compared to the linear function.

The prediction of melting point can be solved using supervised learning where the input feature is the molecular structural information, the target variable is the true melting point of the molecule and the relationship between input and target is given by a regression function.

Regression functions can be linear or non-linear functions. Figure 22 shows the plot of Molecular Weight versus Temperature for hydrocarbons where linear and non-linear function represents the equation $y = ax + b$ and $y = a\log(bx) + c$ respectively. In these equations, a,b, and c are constants, x is the molecular weight and y is the Temperature. We observe the non-linear function is able to approximately fit the data and capture the relationship between molecular weight and temperature compared to the linear function.

This non-linear function is certainly not the perfect fit for the data and can be improved using supervised learning algorithms that can implement non-linear models. Therefore in this section, we will discuss the non-linear models that we can apply to our dataset.

4.1 Support Vector Machines

SVM is a type of supervised learning algorithm that is used for classification or regression analysis as a linear or non-linear model. The steps performed in SVM consist of the separation of the hyperplane, maximizing margins between hyperplane, choosing soft margin, and the kernel function. [43].

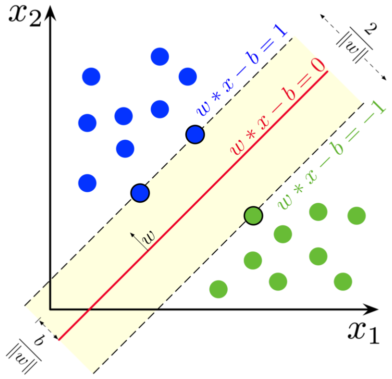


Figure 23: Illustration of hyperplanes in SVM [3]. The red line is the decision boundary and the dashed lines are the margins based on support vectors which are the closest point from each positive class(blue points) and negative class(green points) to the decision boundary.

Suppose we have a training dataset with n samples, input features x_1 and x_2, and the target as a positive and negative class that we want to predict based on the input features. Then the SVM first identifies the decision boundary that can create separation between the positive targets in blue and negative targets in green by creating as shown with a red line in Figure 23. Further, the best decision boundary is chosen based on the maximum separation distance between the positive and negative hyperplanes by selecting the closest support vector points which are the dashed lines representing the margins in the figure. Thus our goal is to maximize the margin. For a dataset with linear separability [44], the margin is given by the equation (20) where $\mathbf{w}$ is weight vector, $\mathbf{x_{pos}}$, and $\mathbf{x_{neg}}$ is x points represented as a vector for positive and negative class. $\frac{2}{||\mathbf{w}||}$ is the margin that we need to maximize.

$$\frac{\mathbf{w^T}(\mathbf{x_{pos}} - \mathbf{x_{neg}})}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||} \tag{20}$$

For non-linear datasets, we perform soft margin classification [44] by including the slack variable ξ and control it using the regularization parameter C as shown in equation (21). Decreasing the value for C will allow for misclassification errors which means x points that lie beyond the support vectors will be allowed.

$$\frac{1}{2}||\mathbf{w}||^2 + C\left(\sum_i \xi^{(i)}\right) \tag{21}$$

$$\mathbf{K}(\mathbf{x^{(i)}}, \mathbf{x^{(j)}}) = \exp\left(-\gamma||\mathbf{x^{(i)}} - \mathbf{x^{(j)}}||^2\right) \tag{22}$$

The non-linear datasets can gain further advantage from the use of the RBF function which projects the existing features onto a higher dimensional plane using the function and then performs the linear separation on the higher dimension and then uses the same function to transform unseen data. The RBF function is given by the equation (22) where $\gamma = \frac{1}{2\sigma^2}$ and σ is the free parameter.

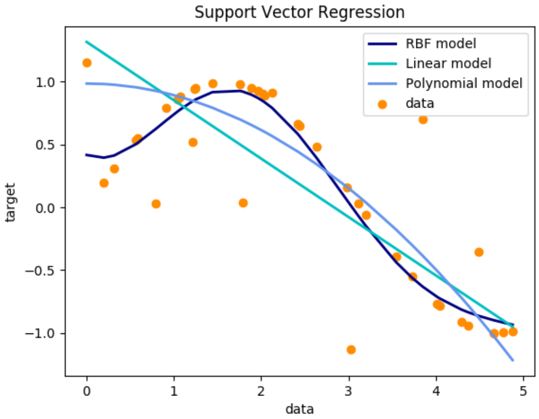


Figure 24: Illustration of regression in SVM. using different types of linear and non-linear kernels [4]. We observe the linear and polynomial model is unable to give an accurate decision line but the RBF model decision line follows the data points more closer than other models.

For a regression problem, the underlying concept is similar to classification but instead of using a decision boundary for positive and negative class separation, the decision boundary is the best-fit regression line that has the maximum number of points as shown in Figure 24 and the output is a continuous variable. Thus the use of soft margin with RBF allows us to create SVM models for non-linear datasets. There are other kernel functions like polynomial kernels but we utilize RBF for its better performance over non-linear data. [45].

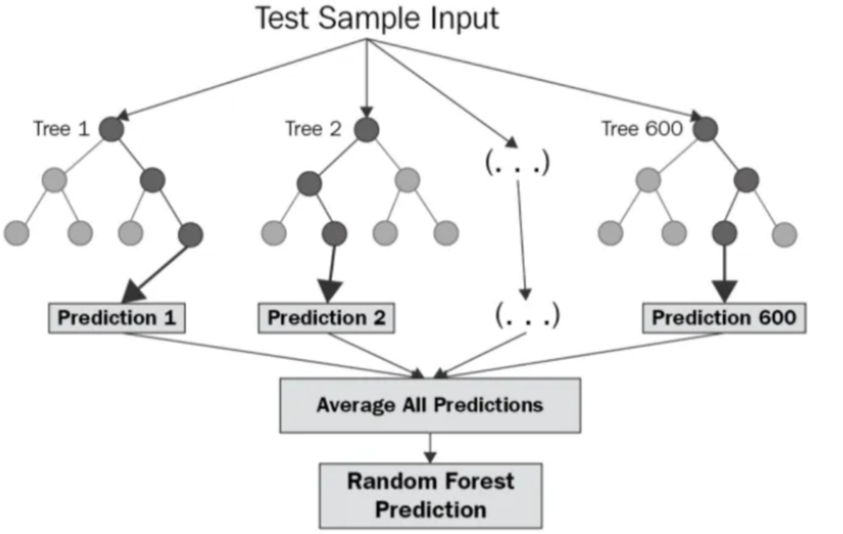


Figure 25: Illustration of Random Forest Regression with 600 decision trees performing prediction on test data. The prediction of each tree is averaged to get the final predicted value [5].

4.2 Random Forests

Random forest is an ensemble learning method for classification and regression problems that create a set of decision trees. For classification, the output is the class selected by the majority voting between trees and for regression, the output is the average prediction of individual trees.

A decision tree creates a tree-like model as shown in Figure 25 where each split is based on maximum information gain. The information gain is calculated based on equation (23) [44] where I is the impurity measure, f is the feature to perform split, D_p and D_j are datasets of parent and jth child node, and N_p and N_j are the total number of samples at the parent and jth child node. The impurity measure is calculated using the equation (24) [44] where $I(t)$ is the impurity measure at node t equal to the mean squared error MSE(t) at node t, N_t is the number of training samples at node t, y^i is the true target value and $\hat{y}_t$ is the predicted target value given by $\frac{1}{N_t}\sum_{i \in D_t}(y^i)$.

$$\text{IG}(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p}(D_j) \tag{23}$$

$$I(t) = \text{MSE}(t) = \frac{1}{N_t}\sum_{i \in D_t}\left(y^i - \hat{y}_t\right)^2 \tag{24}$$

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N}\hat{y}_i \tag{25}$$

Once a set of decision trees are formed, the random forest calculates the predicted target value by the mean or average prediction value of all decision trees given by equation (25) where $\hat{y}$ is the final predicted value, N is the number of trees and $\hat{y}_i$ is the predicted value of each decision tree.

The predicted target value will be the same if all decision trees are given the same data. Thus to avoid overfitting, random forest performs bootstrap aggregating or bagging that selects a random sample with replacement of the training set and fits into the decision trees [46]. This will reduce variance in the model without increasing bias [44].

4.3 AdaBoost

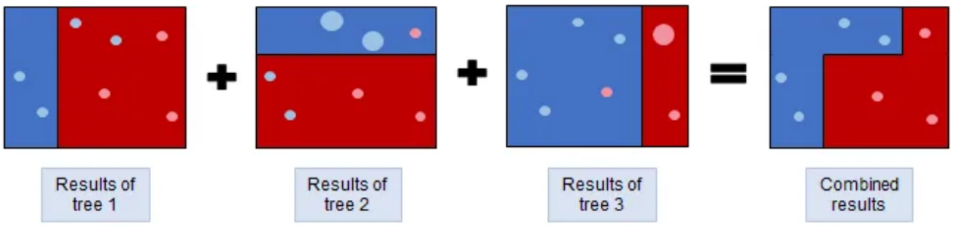


Figure 26: Illustration of Adaboost with 3 decision trees. The final model is a combination of the three decision trees. [6]

AdaBoost [47] is an ensemble ML algorithm that combines several weak learners to make a single strong learner [48]. The algorithm begins with a base estimator model $C_j = \text{train}(\mathbf{X}, \mathbf{y}, \mathbf{w})$ [44] which can be a decision tree or SVR that trains on the dataset. Here the weight $\mathbf{w}$ is distributed uniformly which is equal to $1/N$ where N is the number of samples. Prediction is performed using this weak learner and the weighted error rate is computed as $\epsilon = \mathbf{w} \cdot (\mathbf{y} \neq \hat{\mathbf{y}})$. This is performed to ensure misclassified data gets a higher weight and correctly classified data gets a lower weight so that the next weak learner focuses on correctly classifying the misclassified data. The weights are updated based on the exponential loss function as shown in equation (26) where $\alpha_j = 0.5\log(\frac{1-\epsilon}{\epsilon})$.

$$\mathbf{w} := \mathbf{w} \times \exp(-\alpha_j \times \mathbf{y} \times \hat{\mathbf{y}}) \tag{26}$$

The weights are normalized and re-updated as $\mathbf{w} := \mathbf{w} / \sum_{i=0}^{i=N} w_i$ and the process is again repeated for the next weak learner. The final model combines the output of all weak learners with each weak learner's output weighted according to its performance on training data given by the equation (27) where m is the total number of weak learners.

$$\hat{\mathbf{y}} = \left(\sum_{j=1}^{m} (\alpha_{\mathbf{j}} \times \text{predict}(C_j, \mathbf{X})) \right) \tag{27}$$

AdaBoost is less prone to overfitting than other non-linear regression algorithms because it focuses on misclassified data points and tries to improve their predictions in subsequent iterations, hence creating a more generalized model.

4.4 XGBoost

XGBoost is an ensemble ML algorithm similar to AdaBoost that combines multiple weak learners to create a strong learner and introduces regularization terms to penalize complex models and create more simple models that assist in generalization [49].

In XGBoost the goal is to find the best parameter θ that fits the training data x_i and target y_i where the objective function for θ is given by the equation (28) [49] having $L(\theta)$ as the training loss function and $\Omega(\theta)$ is the regularization function. The Loss function can be given by a Mean Squared Error (MSE) $\sum_i (y_i - \hat{y}_i)^2$ and the regularization function given by $\sum_{i=1}^{t} \omega(f_i)$ having $\omega(f_k)$ as the complexity of tree f_k that helps to avoid overfitting thus penalizing complex trees and encouraging simpler models. Therefore, the objective equation can be updated to equation (29).

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{28}$$

$$obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \omega(f_i) \tag{29}$$

We use L2 regularization as $\Omega(\theta)$ and MSE as $L(\theta)$ and base estimator as DT for training our models.

4.5 Artificial Neural Network

ANN is a type of ML algorithm that mimics the neurons in the human brain. It is widely used in applications like image recognition, speech recognition, natural language processing, etc.

An ANN is composed of an input layer, one or more hidden layers, and an output layer. Each layer is made up of interconnected nodes that are connected to the adjacent layer

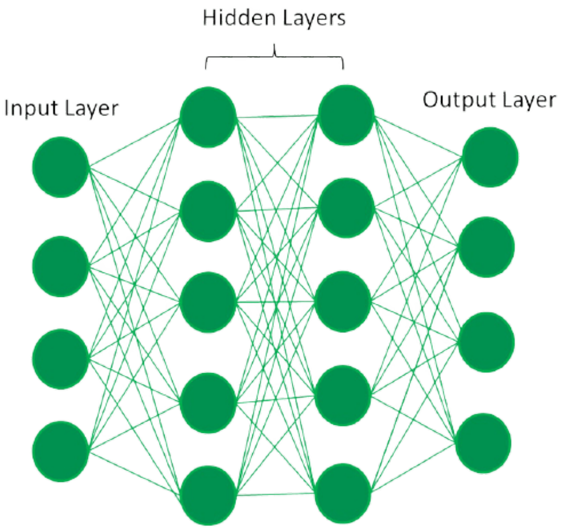


Figure 27: Illustration of ANN with input layer with 4 nodes, 2 hidden layers with 5 nodes, and an output layer with 4 nodes [7].

through weighted connections as shown in Figure 27. The weighted connections enable the network to recognize patterns in the data. The output of each node is calculated by applying a non-linear activation function to the weighted sum of the inputs represented by the equation (30) where $\boldsymbol{w} \cdot \boldsymbol{x}$ represents the dot product of the weight and input vectors, and b is the bias. For regression, the output layer consists of one node to represent the output as a continuous value.

$$y = f(\boldsymbol{w} \cdot \boldsymbol{x} + b) \tag{30}$$

Starting from the input layer, the output of each node is propagated forward to the next layer, where the network detects and recognizes patterns in the data. Based on the network's output, we calculate the error and define a cost function to minimize the error by backpropagating it to the initial layers. Backpropagation allows the network to adjust its weights based on the calculated error, allowing the network to learn from the data and improve its performance over time.

$$f(x) = \max(0, x) \tag{31}$$

For our ANN models we choose Rectified Linear Unit (ReLU) activation function which is a non-linear function as shown in equation (31) where x represents the input to a node. This function outputs the input value x if it is positive, and 0 if negative. This helps us to make the model identify non-linear relationships in the data.

5 Machine Learning Training

In this section, we discuss the steps performed for feature selection, hyperparameter selection, and final model training and testing evaluation method.

5.1 Feature Selection

Feature Selection is the process of selecting a subset of features from a larger set of features available in the dataset. It is performed to prevent overfitting and poor generalization performance of ML models. Overfitting occurs due to excess information making the model more complex, and the parameters keep readjusting to fit the trained model with higher accuracy, leading to lower accuracy in testing. Also, it needs to be taken into consideration that fitting all features is a computationally expensive task.

Scikit-learn provides a range of feature selection methods, out of which we choose the RFE. As stated in scikit-learn [4], "Recursive Feature Elimination is to select features by recursively considering smaller and smaller sets of features." The method trains a base estimator model on the training dataset and ranks each feature according to its importance or coefficients. Feature importance refers to the value assigned to a feature that determines its influence over the predicted target variable. In a tree-based model, the feature importance is calculated based on the feature that has the highest decrease in impurity when it is used for splitting. Based on the feature importance calculated, the least important feature from the training dataset is removed, and the model is retrained on the remaining. These steps are repeated until the stopping criteria are met.

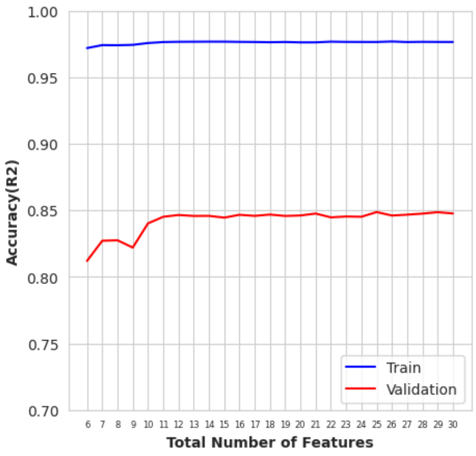


Figure 28: Illustration of an example of RFE plotted against Accuracy(R^2) vs a total number of features available in the dataset. The optimum number of features is chosen as 11.

We use RFE to train the ML model on the training data and perform predictions on validation data. Figure 28 shows an example of RFE applied to the C dataset, which plots the R^2 accuracy score versus the total number of features available. We observe that validation accuracy does not improve after reaching approximately 0.85 with 11 features, and training accuracy is similar for all feature sizes. Therefore, the optimum number of features selected using RFE is 11 for this case. The optimum number of features is chosen based on the first occurrence of the minimum difference between training and validation accuracy.

5.2 Hyperparameter Selection

Hyperparameter selection is the process of choosing optimal hyperparameters for a ML model by providing a set of values and training the model on each set of hyperparameters. The model is evaluated based on its performance on the validation dataset. The set of hyperparameters that give the best performance on the validation dataset is chosen for the final training model. Hyperparameter selection is performed using Grid Search Cross-Validation, which performs a search over specified parameter values on the estimator using cross-validation. Figure 29 shows an illustration of grid search cross-validation.

The Grid Search is performed on the estimator built using a pipeline that follows the order: Standardization, PCA, and the base estimator. Standardization brings all variables to the same scale by having a mean of 0 and a standard deviation of 1 for a feature. PCA is a dimensionality reduction method that transforms the data onto a new coordinate system having each coordinate orthogonal to the other, the first coordinate having the highest variance, the second having the second-highest variance, and so on. The base estimators are classical machine learning algorithms like Support Vector Regressor (SVR), Random-Forests, AdaBoost, and XGBoost. The pipeline is provided with different combinations of hyperparameters set as shown in Figure 29 represented by Step A. Each hyperparameter set undergoes cross-validation to evaluate its performance.

Cross-validation splits the datasets into multiple parts or folds where 1 fold is the validation dataset and the rest is the training dataset. The ML model is trained on the training fold, and its performance is validated on the validation fold, as shown in Figure 29 with Step B. This process is repeated multiple times and the average performance of training and validation results is evaluated as shown in Step C. It helps to present an accurate representation of the model that is evaluated from the mean and standard deviation of the performance of training and validation, which allows us to view the overfitting and underfitting of the data by the model. Here, cross-validation is performed using a 5 split KFold that divides the dataset into 5 equally sized folds, where one fold acts as the validation set and 4 folds as the training set, thus having the train-to-validation ratio equal to $80 : 20$. The R^2 score is used to evaluate the performance.

The best parameters reported by grid search cross-validation are not always the best results since it reports the highest-scoring model, which can also be an overfit model. Therefore, we define the below criteria to choose the best parameters:

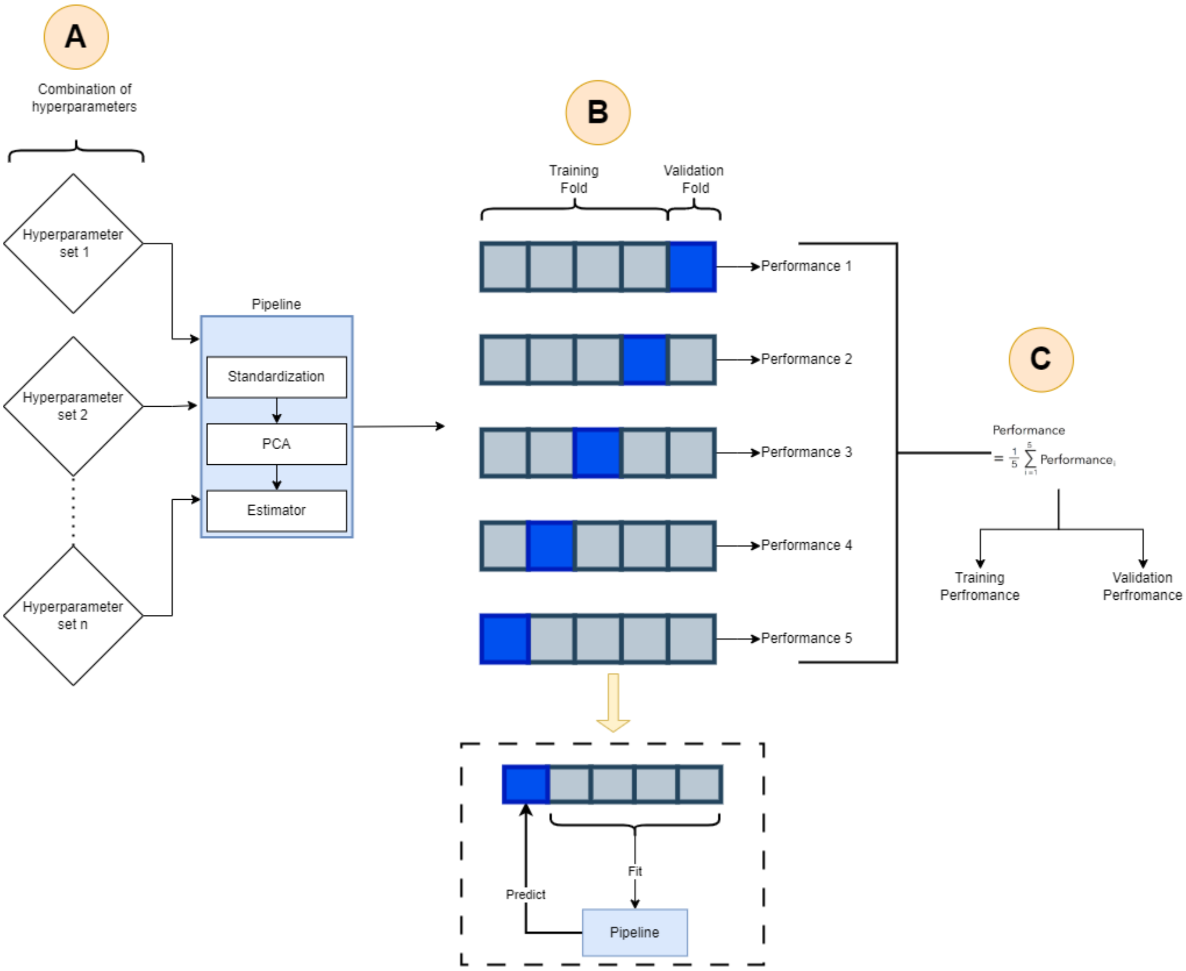


Figure 29: Illustration of Hyperparameter Selection using Grid Search Cross-Validation. (A) represents grid search where different combinations of hyperparameter sets are given to the pipeline. (B) represents cross-validation where each hyperparameter set is used to train a pipeline on the training fold and prediction is performed on the validation fold. 5 Fold cross-validation is applied. (C) represents the average performance of cross-validation of each hyperparameter set that results in training and validation performance scores.

- We select the model parameters with the highest mean validation score.
- We check for other model parameters that have ±0.03 deviations from the highest mean validation score.
- We choose the model parameters having the lowest difference between the mean train score and the mean validation score.

5.3 Training, Validation and Testing

Figure 30 represents the process used for model training, validation, and testing. The steps are as follows:

- The dataset is split into 5 folds, with 1 fold representing test data and 4 folds representing the training dataset. This is represented as Step A.
- Each training dataset from the previous split, is further split into 5 folds, where 1 fold represents validation data and 4 folds represent train data. The pipeline is trained on train data, and prediction is performed on validation data. This is represented as Step B.
- The pipeline is retrained on both train and validation data, and prediction is performed on test data. It is represented as Step C.
- Steps B and C are repeated for all splits from Step A.
- The final result is the average performance score of training, validation, and test data.

Steps A and C involve splitting data into train and test and validating on different test data, which is a necessary step because each split shuffles the data, which results in imbalanced train and test data. It means there can be a split where test data may have all the high temperatures and the model learns from the low-temperature molecules only, which will result in poor performance in the prediction of the test. Thus, a good evaluation of the ML model will be based on the average score of multiple test folds.

5.4 Performance Metrics

Coefficient of determination (R^2) and RMSE are used as metrics to evaluate the performance of models. R^2 is used in feature selection and hyperparameter selection and as the evaluation index for comparing train, validation, and test accuracy in terms of regression. RMSE is only used in the final evaluation of train, validation, and test data discussed in Section 5.3.

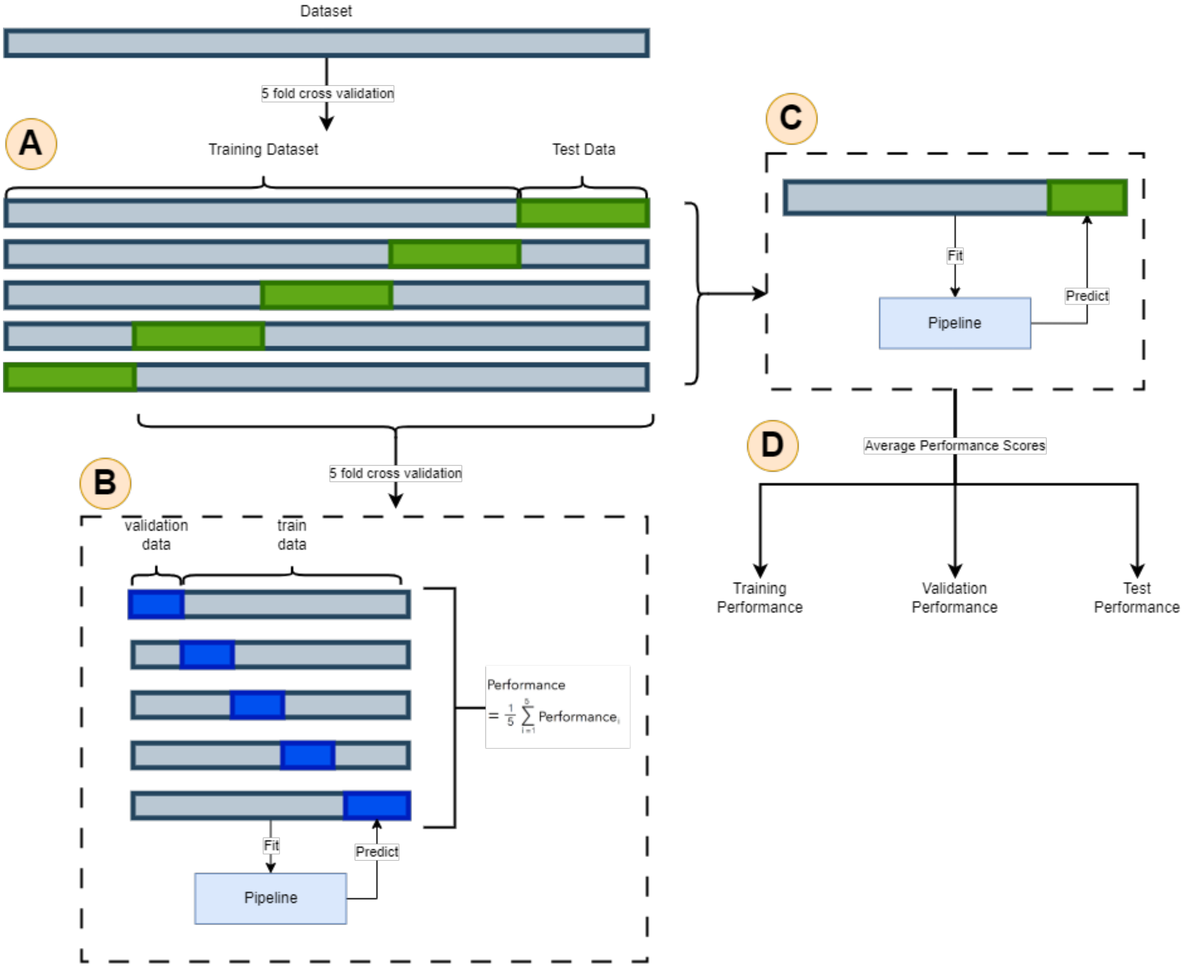


Figure 30: Illustration of Model Training Validation and Testing. (A) The dataset is split into 5 folds with 1 fold as test data and 4 folds as the training dataset. (B) The training dataset is split into 5 folds with 1 fold as validation data and the rest as train data. Prediction is performed on Validation data. (C) Prediction is performed on Test data folds resulting from Step A. (D) Final evaluation metrics provide the performance of train, validation, and test data.

RMSE

RMSE is the square root of the mean value of Sum of Squared Error (SSE) where the error is the difference between a true value and a predicted value. RMSE is calculated using the equation (32) [44] where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the total number of samples. It measures the average distance of a predicted value from a true value.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{\text{SSE}}{n}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{32}$$

RMSE with value 0 represents that the predicted values are equal to the true values for n samples and the model is 100% accurate. When the value is large it means predicted values are not closer to the true values and the model is not accurate. Also if RMSE of train and validation is smaller and the test data is larger, then the model is overfit and if train and validation are larger and the test data is smaller, then the model is underfit.

R^2

R^2 score measures how well the model fits the data. It measures if the proportion of variability in the true value is being accounted for by the regression model. R^2 is calculated using the equation (33) [44] where Total sum of squares (SST) is $\sum_i (y_i - \mu)$ which represents the variance of the response.

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\text{MSE}}{\text{Var}(y)} \tag{33}$$

R^2 score ranges from 0 to 1, where 0 represents that the model does not explain any variability in the data and 1 represents that the model explains all variability in the data. A high R^2 for train and validation data but a low for test data represents overfitting. A low R^2 for train and validation data but a high for test data represents underfitting.

6 Results

The results comprise of the ML model discussed in Section 4 trained on the C, CX, CXOS, and CXOSNP datasets discussed in Section 2.4. The datasets are split into the ratio train-to-validation-to-test $= 60 : 20 : 20$. Cross-validation is performed using KFold with 5 splits, as discussed in Section 5.3. The descriptive features discussed in Table 2 comprise 97 features that are extracted from each dataset. In each dataset, the descriptive feature with a value of zero in all rows is eliminated. The remaining features are trained on ANN, SVR, Random Forest, XGBoost, and AdaBoost which use base estimators such as DT and SVR.

Morgan Fingerprint and Mol2Vec embedding features are trained on ANN. We attempted to train them on Random Forest, SVR, AdaBoost but the R^2 scores were negative and their performance was poor. We also attempted on XGBoost whose performance was good but not as superior as ANN. The shape of Morgan Fingerprint is created using a radius of 1 and an array size of 256. The Mol2Vec algorithm creates a feature with a fixed array size of 300.

The results of Morgan Fingerprint and Mol2Vec are used for comparison with our descriptive features, and we attempt to achieve lower RMSE for the descriptive features compared to them.

For training and validation data, we compare using R^2 score, and test data comparison is performed using RMSE score. With the increasing size of the dataset, the variance in temperature is increasing, and the test R^2 score is decreasing. The test RMSE gives the absolute error value of the melting point temperature in Kelvin.

C Dataset

The C Dataset has 22 features, 1526 molecules, and their melting point temperature in Kelvin. The data is split into train, validation, and test sizes of 977, 244, and 305 respectively. All descriptive features undergo standardization.

Below are the ML model hyperparameters used for training and prediction.

- Random Forest uses RFE with a base estimator of Random Forest with 50 estimators and other default values for feature selection, and we select 11 features. PCA is not performed since it reduces the final test accuracy. The hyperparameters for Random Forest are 30 estimators, 10 max depth, and other default values.
- AdaBoost(DT) uses RFE with a base estimator of AdaBoost with 50 DT and other default values for feature selection, and we select 13 features. PCA is not performed since it reduces the final test accuracy. The hyperparameters of AdaBoost(DT) are

30 estimators, exponential loss, 0.1 learning rate, and the base estimator is a DT with a max depth of 8.

- SVR does not use RFE and performs feature selection using PCA, and we select 11 PCA components. The hyperparameters are RBF kernel, C = 100.0, and gamma = 0.01.

- AdaBoost(SVR) does not uses RFE, and performs feature selection using PCA, and we select 11 PCA components. The hyperparameters of AdaBoost are 20 estimators, exponential loss, 0.1 learning rate, and the base estimator is a SVR with C = 100.0 and gamma = 0.1.

- XGBoost uses RFE with a base estimator of XGBRegressor with objective as squared error and booster as gradient boost tree and other default values. We selected 8 features. The hyperparameters are objective as squared error and booster as gradient boost tree, 0.2 learning rate, lambda of 1.2 as L2 regularization, 7 max depth, 30 estimators, and evaluation metrics as RMSE.

- ANN does not use RFE but performs feature selection using PCA and we select 12 PCA components. The ANN has an input layer of shape 12, the first hidden layer with 256 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 100 with 20 epochs.

- Morgan FingerPrint's ANN model has an input layer of shape 256, the first hidden layer with 1024 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 100 with 20 epochs.

- Mol2Vec's ANN model has an input layer of shape 300, the first hidden layer with 1024 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 100 with 20 epochs.

Table 3 shows the R^2 score and RMSE for ML models applied to descriptive features, Morgan Fingerprint, and Mol2Vec. We observe in descriptive features, Random Forest has the highest R^2 of 0.81 and the lowest RMSE of 51.00 among all other models. It also performed better than the ANN model with Morgan Fingerprint and Mol2Vec. For the C dataset, our descriptive features can perform better than the embedding features.

Even though Random Forest has the best test scores, it is not the most generalized model since the train and validation R^2 are 0.96 and 0.90 which is slightly overfitting. We observe AdaBoost(SVR) has a train and validation R^2 of 0.81 and 0.80, which is a more generalized model compared to Random Forest even though its test RMSE is 53.34, which is +2 higher than RandomForest.Also, SVR has a generalized model with train and validation R^2 of 0.80 and 0.79 respectively, and test RMSE of 53.87. XGBoost and ANN

Type of Feature	Model	Train		Validation		Test	
		R^2	RMSE	R^2	RMSE	R^2	RMSE
Descriptive Features	Random Forest	0.96 ± 0.00	23.98 ± 0.72	0.90 ± 0.03	37.11 ± 5.58	0.81 ± 0.02	51.00 ± 3.03
	AdaBoost(DT)	0.96 ± 0.00	23.84 ± 0.74	0.90 ± 0.03	37.96 ± 5.59	0.81 ± 0.02	51.46 ± 3.09
	SVR	0.80 ± 0.00	51.97 ± 1.08	0.79 ± 0.03	53.96 ± 4.67	0.79 ± 0.03	53.87 ± 3.76
	AdaBoost(SVR)	0.81 ± 0.00	51.01 ± 1.08	0.80 ± 0.03	53.13 ± 4.51	0.79 ± 0.03	53.34 ± 3.98
	XGBoost	0.94 ± 0.01	28.49 ± 3.21	0.88 ± 0.03	40.31 ± 6.08	0.80 ± 0.02	52.27 ± 3.30
	ANN	0.93 ± 0.01	31.11 ± 2.54	0.89 ± 0.02	40.62 ± 4.10	0.80 ± 0.01	56.11 ± 3.71
Morgan Fingerprint	ANN	0.93 ± 0.01	31.91 ± 3.70	0.89 ± 0.02	41.22 ± 7.73	0.80 ± 0.01	59.60 ± 15.25
Mol2Vec	ANN	0.83 ± 0.06	50.80 ± 12.87	0.80 ± 0.08	56.66 ± 14.77	0.77 ± 0.05	58.33 ± 8.30

Table 3: Train, Validation and Test R^2 score and RMSE for C dataset

models are also overfitting due to high train and validation R^2 scores, and test RMSE of 52.27 and 56.11 respectively.

From these observations, AdaBoost(SVR) is the most generalized model, and Random Forest has the highest test RMSE.

CX Dataset

The CX Dataset has 32 features, 2825 molecules, and their melting point temperature in Kelvin. The data is split into a train, validation, and test sizes of 1808, 452, and 565, respectively. All features undergo standardization.

Below are the ML model hyperparameters used for training and prediction.

- Random Forest uses RFE with a base estimator of Random Forest with 50 estimators and other default values for feature selection, and we select 12 features. PCA is not performed since it reduces the final test accuracy. The hyperparameters for Random Forest are 30 estimators, 15 max depth, and other default values.
- AdaBoost(DT) uses RFE with a base estimator of AdaBoost with 50 DT and other default values for feature selection, and we select 14 features. PCA is not performed since it reduces the final test accuracy. The hyperparameters of AdaBoost(DT) are 30 estimators, exponential loss, 0.1 learning rate, and the base estimator is an DT with a max depth of 8.
- SVR does not use RFE and performs feature selection using PCA and we select 15

PCA components. The hyperparameters are RBF kernel, C = 1000.0, and gamma = 0.01.

- AdaBoost(SVR) does not use RFE and performs feature selection using PCA and we select 15 PCA components. The hyperparameters of AdaBoost are 20 estimators, exponential loss, 0.1 learning rate, and the base estimator is a SVR with C = 1000.0 and gamma = 0.1.
- XGBoost uses RFE with a base estimator of XGBRegressor with objective as squared error and booster as gradient boost tree and other default values. We selected 8 features. The hyperparameters are objective as squared error and booster as gradient boost tree, 0.2 learning rate, lambda of 1.5 as L2 regularization, 7 max depth, 30 estimators, and evaluation metrics as RMSE.
- ANN does not use RFE but performs feature selection using PCA and we select 15 PCA components. The ANN has an input layer of shape 15, the first hidden layer with 256 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 200 with 30 epochs.
- Morgan FingerPrint's ANN model has an input layer of shape 256, the first hidden layer with 1024 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 200 with 30 epochs.
- Mol2Vec's ANN model has an input layer of shape 300, the first hidden layer with 1024 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 200 with 30 epochs.

Table 4 shows the R^2 score and RMSE for ML models applied on descriptive features, Morgan Fingerprint, and Mol2Vec. We observe in descriptive features, Random Forest has the highest R^2 of 0.81 and the lowest RMSE of 47.80 among all other models. It also performed lower than the ANN model trained with Morgan Fingerprint which has R^2 of 0.88 and the lowest RMSE of 39.42. For the CX dataset, our descriptive features are not able to perform better than the Morgan Fingerprint.

Random Forest has the best test scores among descriptive features with train and validation R^2 of 0.96 and 0.91, which is overfitting. SVR and AdaBoost(SVR) are generalized models with train and test R^2 scores of 0.84 and 0.80, respectively, and AdaBoost(SVR) has a better test RMSE of 49.59. XGBoost is trained with only 8 features and is an overfit model, but the test RMSE is closer to other models. ANN model also performed train and test R^2 score of 0.86 and 0.81 which is overfitting but the RMSE value is 49.25 which is closer to other models

From these observations, SVR and AdaBoost(SVR) are the most generalized models, and Random Forest has the highest test RMSE but the descriptive features were not able to surpass the Morgan Fingerprint ANN model in test RMSE.

Type of Feature	Model	Train		Validation		Test	
		R^2	RMSE	R^2	RMSE	R^2	RMSE
Descriptive Features	Random Forest	0.96 ± 0.00	20.25 ± 0.53	0.91 ± 0.03	33.91 ± 5.44	0.81 ± 0.03	47.80 ± 2.12
	AdaBoost(DT)	0.91 ± 0.00	32.51 ± 0.68	0.87 ± 0.02	40.47 ± 3.94	0.79 ± 0.02	49.79 ± 2.32
	SVR	0.84 ± 0.00	45.02 ± 0.98	0.82 ± 0.02	48.24 ± 3.31	0.80 ± 0.02	50.37 ± 2.10
	AdaBoost(SVR)	0.84 ± 0.00	43.71 ± 0.93	0.83 ± 0.02	47.07 ± 3.18	0.80 ± 0.02	49.59 ± 2.03
	XGBoost	0.91 ± 0.00	32.82 ± 1.23	0.86 ± 0.02	41.37 ± 4.15	0.78 ± 0.03	51.66 ± 2.77
	ANN	0.86 ± 0.02	41.37 ± 2.55	0.84 ± 0.04	45.11 ± 3.62	0.81 ± 0.02	49.25 ± 2.37
Morgan Fingerprint	ANN	0.94 ± 0.00	27.29 ± 1.77	0.92 ± 0.01	31.66 ± 2.63	0.88 ± 0.01	39.42 ± 3.15
Mol2Vec	ANN	0.76 ± 0.12	52.86 ± 10.06	0.75 ± 0.11	54.97 ± 10.46	0.80 ± 0.01	49.14 ± 1.40

Table 4: Train, Validation and Test R^2 score and RMSE for CX dataset

CXOS Dataset

The CXOS Dataset has 53 features, 24265 molecules, and their melting point temperature in Kelvin. The data is split into a train, validation, and test sizes of 15529, 4853, and 3883, respectively. All features undergo standardization.

Below are the ML model hyperparameters used for training and prediction.

- Random Forest uses RFE with a base estimator of Random Forest with 50 estimators and other default values for feature selection, and we select 26 features. PCA is not performed since it reduces the final test accuracy. The hyperparameters for Random Forest are 30 estimators, 20 max depth, and other default values.
- AdaBoost(DT) uses RFE with a base estimator as AdaBoost with 50 Decision Trees and other default values for feature selection, and we select 21 features. PCA is not performed since it reduces the final test accuracy. The hyperparameters of AdaBoost(DT) are 30 estimators, exponential loss, 1.0 learning rate, and the base estimator is a DT with a max depth of 15.
- SVR does not use RFE and performs feature selection using PCA and we select 31 PCA components. The hyperparameters are RBF kernel, C = 1000.0, and gamma = 0.01.
- AdaBoost(SVR) does not use RFE and performs feature selection using PCA and we select 31 PCA components. The hyperparameters of AdaBoost are 20 estimators, exponential loss, 1.0 learning rate, and the base estimator is a SVR with C = 1000.0 and gamma = 0.01.

- XGBoost uses RFE with a base estimator of XGBRegressor with objective as squared error and booster as gradient boost tree and other default values. We selected 22 features. The hyperparameters are objective as "reg: squarederror", booster as "gbtree", 0.2 learning rate, lambda 1.5 as L2 regularization, 7 max depth, 30 estimators, and evaluation metrics as RMSE.

- ANN does not use RFE but performs feature selection using PCA, and we select 31 PCA components. The ANN has an input layer of shape 31, the first hidden layer with 256 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 200 with 40 epochs.

- Morgan FingerPrint's ANN model has an input layer of shape 256, the first hidden layer with 1024 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 500 with 30 epochs.

- Mol2Vec's ANN model has an input layer of shape 300, the first hidden layer with 1024 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 500 with 30 epochs.

Type of Feature	**Model**	**Train**		**Validation**		**Test**	
		R^2	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**
Descriptive Features	Random Forest	0.93 ± 0.00	19.54 ± 0.28	0.79 ± 0.04	29.73 ± 3.01	0.64 ± 0.01	41.71 ± 0.63
	AdaBoost(DT)	0.95 ± 0.00	16.64 ± 0.88	0.81 ± 0.05	29.08 ± 3.61	0.63 ± 0.02	42.45 ± 0.55
	SVR	0.76 ± 0.00	36.26 ± 0.18	0.67 ± 0.02	38.39 ± 1.05	0.67 ± 0.01	41.99 ± 0.47
	AdaBoost(SVR)	0.70 ± 0.01	38.19 ± 0.52	0.59 ± 0.02	40.92 ± 0.94	0.55 ± 0.00	46.70 ± 0.23
	XGBoost	0.74 ± 0.00	35.71 ± 0.56	0.64 ± 0.02	38.09 ± 0.98	0.64 ± 0.01	41.60 ± 0.54
	ANN	0.72 ± 0.03	37.94 ± 1.60	0.63 ± 0.04	39.84 ± 2.01	0.64 ± 3.25	44.59 ± 3.25
Morgan Fingerprint	ANN	0.96 ± 0.02	15.45 ± 3.70	0.89 ± 0.08	22.42 ± 8.40	0.70 ± 0.18	40.93 ± 11.64
Mol2Vec	ANN	0.43 ± 0.24	49.70 ± 5.41	0.31 ± 0.30	49.16 ± 5.64	0.55 ± 0.04	46.24 ± 0.89

Table 5: Train, Validation and Test R^2 score and RMSE for CX dataset

Table 5 shows the R^2 score and RMSE for ML models applied on descriptive features, Morgan Fingerprint, and Mol2Vec. We observe in descriptive features, XGBoost has a test R^2 of 0.64 and the lowest RMSE of 41.60 among all other models, and its performance is the best. At first glance, ANN model trained with Morgan Fingerprint seems to have the lowest test RMSE of 40.94, but the standard deviation is ±11.64 which means the model performance is poor with different train and test folds. Mol2Vec performed better with a test RMSE of 46.24 with a standard deviation of ±0.89 which is better than Morgan Fingerprint. For the CXOS dataset, XGBoost performance is superior to Mol2Vec's ANN model, and we can perform better than embedding features.

Random Forest, SVR, and XGBoost have similar test RMSE values, but SVR, and XGBoost are more generalized models compared to Random Forest, with a high train R^2 and very low RMSE of 0.93 and 19.54, respectively. Among all descriptive feature models AdaBoost performed worst with a test RMSE of 41.60, which is the highest RMSE among all models.

From these observations, SVR and XGBoost are the most generalized models and can surpass the Mol2Vec ANN model's performance.

CXOSNP Dataset

The CXOSNP Dataset has 97 features, 65466 molecules, and their melting point temperature in Kelvin. The data is split into train, validation, and test sizes of 41897, 10475, and 13094 respectively. All features undergo standardization. Below are the ML model hyperparameters used for training and prediction.

- Random Forest uses RFE with a base estimator of Random Forest with 50 estimators and other default values for feature selection and we select 31 features. PCA is not performed since it reduces the final test accuracy. The hyperparameters for Random forest are 40 estimators, 20 max depth, and other default values.
- AdaBoost(DT) uses RFE with base estimator as AdaBoost with 50 DT and other default values for feature selection and we select 27 features. PCA is not performed since it reduces the final test accuracy. The hyperparameters of AdaBoost(DT) are 30 estimators, exponential loss, 0.1 learning rate, and the base estimator is a DT with a max depth of 20.
- SVR does not uses RFE and performs feature selection using PCA and we select 61 PCA components. The hyperparameters are RBF kernel, C = 1000.0, and gamma = 0.01.
- AdaBoost(SVR) does not uses RFE and performs feature selection using PCA and we select 61 PCA components. The hyperparameters of AdaBoost are 20 estimators, exponential loss, 0.1 learning rate, and the base estimator is a SVR with C=100.0 and gamma=0.1.
- XGBoost uses RFE with base estimator as XGBRegressor with objective as squared error and booster as gradient boost tree and other default values. We select 31 features. The hyperparameters are objective as squared error, booster as gradient boost tree, 0.2 learning rate, lambda 1.0 as L2 regularization, 15 max depth, 30 estimators, and evaluation metrics as RMSE.

- ANN does not use RFE but performs feature selection using PCA and we select 68 PCA components. The ANN has an input layer of shape 15, the first hidden layer with 256 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and Adam optimizer with a 0.1 learning rate. The batch size is 1000 with 30 epochs.
- Morgan FingerPrint's ANN model has an input layer of shape 256, the first hidden layer with 1024 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 2000 with 35 epochs.
- Mol2Vec's ANN model has an input layer of shape 300, the first hidden layer with 1024 nodes, the second hidden layer with 64 nodes, and the output layer with 1 node. It uses MSE loss and an Adam optimizer with a 0.1 learning rate. The batch size is 2000 with 40 epochs.

Type of Feature	**Model**	**Train**		**Validation**		**Test**	
		R^2	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**
Descriptive Features	Random Forest	0.88 ± 0.00	21.56 ± 0.10	0.66 ± 0.05	30.60 ± 2.41	0.42 ± 0.05	42.66 ± 2.73
	AdaBoost(DT)	0.96 ± 0.00	13.63 ± 0.93	0.42 ± 0.03	44.85 ± 2.83	0.40 ± 0.08	44.27 ± 2.62
	SVR	0.73 ± 0.01	33.30 ± 1.05	0.51 ± 0.03	44.62 ± 1.28	0.43 ± 0.05	43.08 ± 2.89
	AdaBoost(SVR)	0.76 ± 0.01	31.45 ± 0.27	0.51 ± 0.02	42.83 ± 0.61	0.43 ± 0.07	43.70 ± 2.57
	XGBoost	0.87 ± 0.01	22.45 ± 1.18	0.68 ± 0.05	31.03 ± 2.58	0.48 ± 0.06	42.63 ± 2.36
	ANN	0.59 ± 0.05	38.88 ± 2.33	0.44 ± 0.06	40.28 ± 2.38	0.41 ± 0.08	46.11 ± 1.20
Morgan Fingerprint	ANN	0.94 ± 0.06	15.56 ± 5.95	0.82 ± 0.15	24.72 ± 0.30	0.59 ± 0.19	44.96 ± 10.53
Mol2Vec	ANN	0.40 ± 0.10	43.86 ± 2.18	0.24 ± 0.13	43.28 ± 1.96	0.37 ± 0.04	45.56 ± 1.47

Table 6: Train, Validation and Test R^2 score and RMSE for CXOSNP dataset

Table 6 shows the R^2 score and RMSE for ML models applied on descriptive features, Morgan Fingerprint, and Mol2Vec for the CXOSNP dataset. We observe in descriptive features, XGBoost has the lowest RMSE of 42.63 among all other models, and its test performance is the best. But the train R^2 score is 0.87, which is an overfit model since the validation and test R^2 scores are 0.68 and 0.48, respectively.

Among the ANN models for Morgan Fingerprint and Mol2Vec, Morgan Fingerprint has a lower test RMSE of 44.96, but the standard deviation is $pm10.53$, which is a wide range. It means the model does not perform well with different test folds. But the Mol2Vec has a test RMSE of 45.65 with a standard deviation of $pm1.47$ which is lower than the Morgan Fingerprint. The Mol2Vec model can generalize better since the train and test R2 scores are 0.40 and 0.37, respectively. Therefore, among embedding features, Mol2Vec performs better.

All models exceptANN perform relatively similarly for descriptive features, and all of them are overfitting. ANN model's performance test RMSE is the highest at 46.11, but the model is more generalized since train and test R^2 have a difference of 0.10, which is less compared to other descriptive models.

From the observations of all datasets, we can conclude XGBoost and Random Forest usually overfit, but their test RMSE values are the lowest. ANN and SVR are the most generalized models that can be used, even though their test RMSE values are not the lowest. Among embedding features, the Mol2Vec ANN model is more generalized and gives a stable output compared to Morgan Fingerprint.

Also, we can conclude that our descriptive features can perform better than embedding features, but they are not superior to them, as seen in the results of the CXOSNP dataset.

7 Discussion

ML models can be analyzed using model-specific strategy and model-agnostic strategy. In model-specific strategy, we analyze the reasons for the performance of the models using feature selection and principal component analysis. We attempted a model-agnostic strategy using partial dependence plots but these plots consider each feature independent of each other, which cannot be considered for molecules since the features are dependent. Therefore, we only analyze based on a model-specific strategy which provides us the goodness of the features we create and helps us to understand which features have importance.

Further, we discuss about the improvement that can be made in feature engineering.

7.1 Model Analysis

Random Forest

Figure 31 shows the feature importance scores of each feature in all the datasets for the Random Forest. We select 11, 12, 26, and 31 features for the C, CX, CXOS, and CXOSNP datasets respectively.

In the C dataset shown in Figure 31a, the Random Forest gives the highest importance to total bonds and rings. It is followed by density representing compactness, eccentricity representing flatness, sphericity representing the spherical shape, and rotatable bonds representing flexibility. Balaban J Index which describes the structural topology of a molecule represented as a graphical index number, is also of importance. Further, the selection includes volume, molecular weight, and single bonds.

Therefore, we observe that the model chooses the shape of the molecule using bonds and rings and further requires physical features like density, eccentricity, sphericity, rotatable bonds, and volume to predict the melting point. It does not need electronegativity since all atoms are carbon and have the same electronegativity values. It does not select double and triple bonds because the information is available in Total Bonds. We also observe that it is trying to identify aromaticity using total bonds, total rings, and aromatic bonds, which are the top three features.

In the CX dataset shown in Figure 31b, the highest feature importance score is for volume. All features selected in the C dataset are also selected in the CX dataset. Further, the electronegativity difference mean is also included. Since the model chooses volume as the most important feature, we can infer that the information about halogens is captured by the increase in the size of the molecule and is represented through volume. Therefore, it does not include the count of halogen atoms. Compared to the C dataset, molecular weight has higher importance since this feature can capture information about halogens.

The electronegativity difference mean is also selected, even though its feature importance score is lower. It means that among the electronegativity features we created, the difference in electronegativity between the adjacent atoms captures information better than other equations for this dataset.

In the CXOS dataset, as shown in Figure 31c, total bonds and rings have again become the most important feature, similar to Figure 31a for C Dataset. It can be explained by the fact that the ratio of hydrocarbons to halogen compounds to halogen, oxygen, and sulfur compounds is 6 : 5 : 89 in the CXOS dataset, due to which the model does not consider volume as an important feature since the data has a higher ratio of oxygen and sulfur compounds.

The model selects four types of electronegativity equations: electronegativity mean, electronegativity variance, electronegativity variance mean, electronegativity variance variance, and electronegativity difference variance. Since the halogen atom count has not been selected, it is assumed that information about halogens is captured through these features. And it may also capture the effects of oxygen's and sulfur's electronegativity.

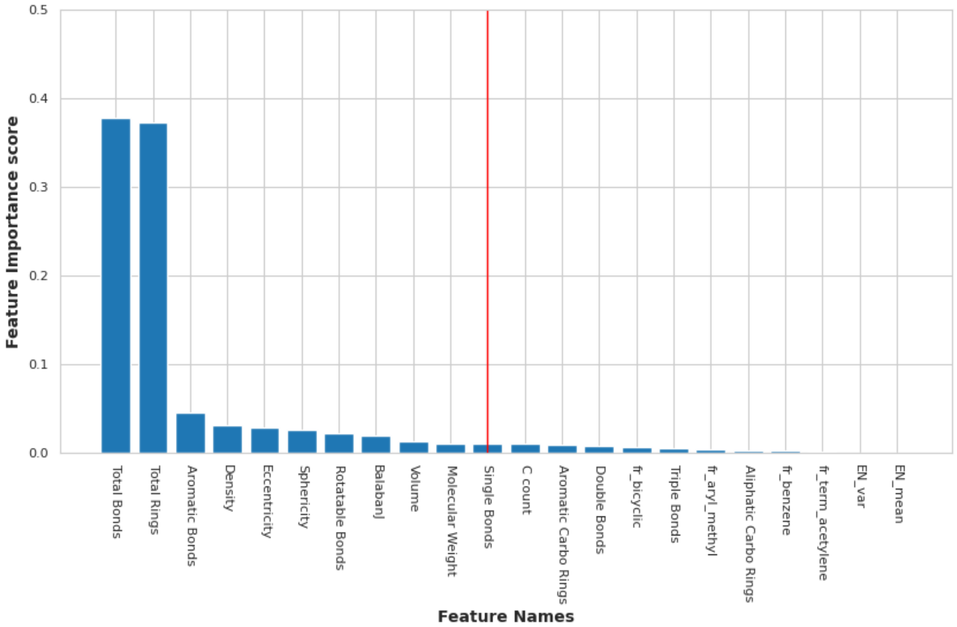
Further functional groups containing oxygen, like aliphatic and aromatic carboxyl, aliphatic and aromatic hydroxyl, and ether are selected. It means the model can capture information on hydrogen bonding through these functional groups. If we are to state that the model chooses only oxygen-related functional groups over sulfur since carboxyl and hydroxyl groups have stronger hydrogen bonds compared to sulfur compounds then we can observe that the ratio of compounds containing at least one oxygen to that of compounds containing at least one sulfur in the CXOS dataset is 85 : 15. It means there are fewer compounds to represent sulfur information, and therefore its feature importance score is lower.

In the CXOSNP dataset shown in Figure 31d, total rings have the highest feature importance score. The model chooses the nitrogen count since there are 40,156 molecules with at least one nitrogen, which represents 61% of the molecules in the CXOSNP dataset. Therefore, it has a high feature importance score.

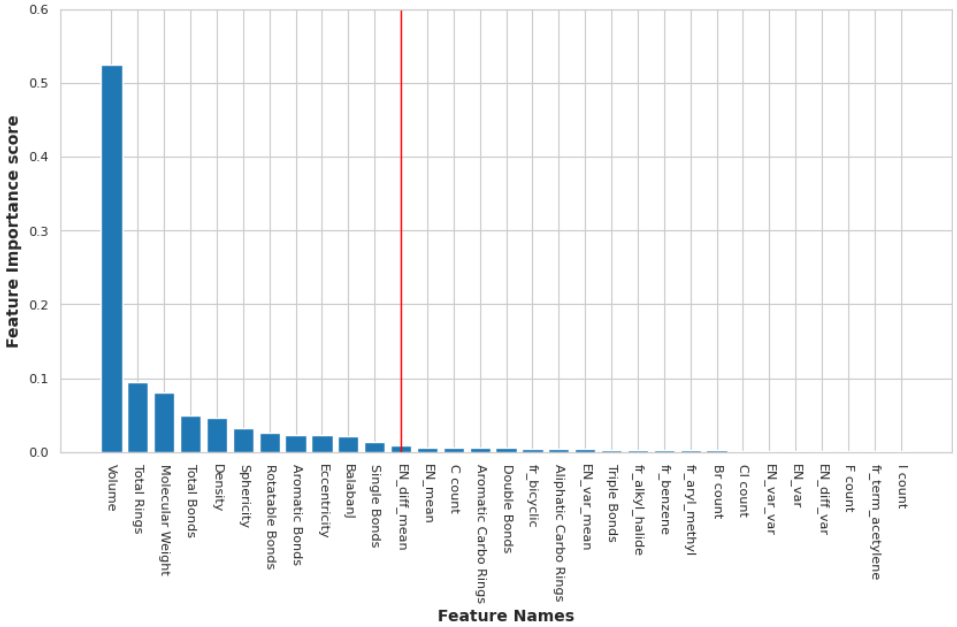
Similar to the results in Figure 31a, the model chooses density, total bonds, Balaban J Index, eccentricity, sphericity, molecular weight, and rotatable bonds to identify the features that can describe the shape of the molecule. It shows that the melting point has a stronger correlation with the shape of the molecule. Hence, these features are getting higher feature importance scores.

We also observe that many functional groups like amine, aliphatic and aromatic carboxyl, aliphatic and aromatic hydroxyl, ether, and ketone are selected. These features are built for identifying hydrogen bonding, which shows the model finds a correlation between hydrogen bonding and melting point and requires these features to identify their presence in the molecule for prediction.

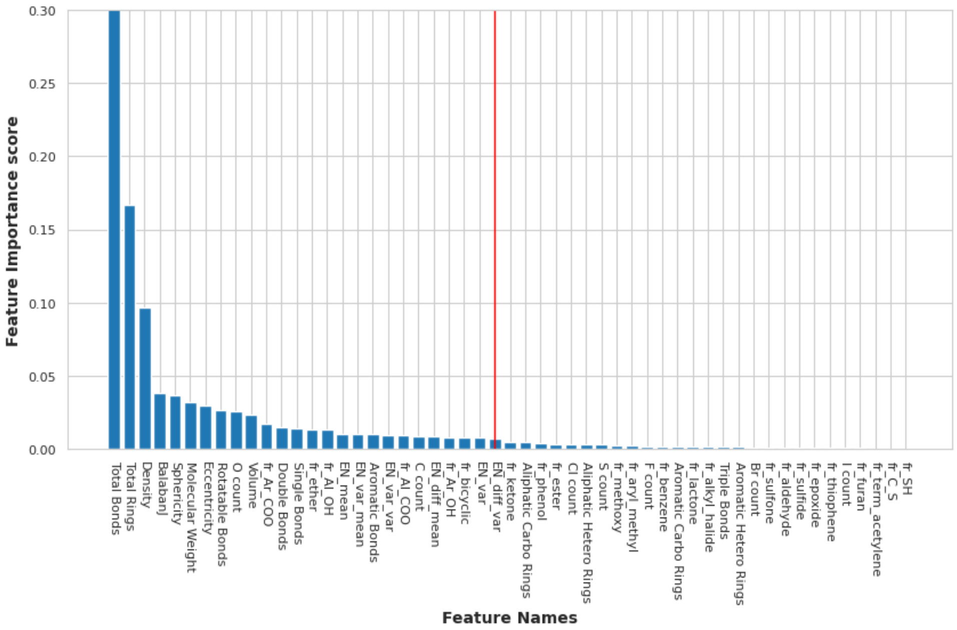
Based on the analysis, we can conclude that the Random Forest model considers physical shape descriptive features to be the most important feature for predicting melting points.



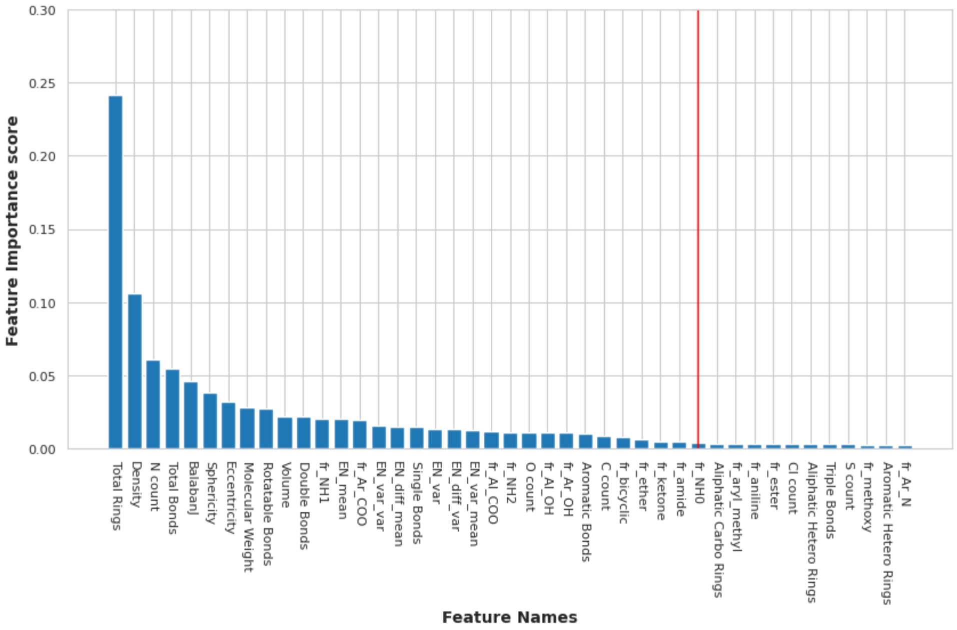
(a) Plot of Feature Importance Scores on C Dataset for Random Forest. 11 features are chosen.



(b) Plot of Feature Importance Scores on CX Dataset for Random Forest. 12 features are chosen.



(c) Plot of Feature Importance Scores on CXOS Dataset for Random Forest. 26 features are chosen.



(d) Plot of Feature Importance Scores on CXOSNP Dataset for Random Forest. Only the first 38 features are shown in descending order of feature score. 31 features are chosen.

Figure 31: Plots of Feature Importance scores for Random Forest on C, CX, CXOS, and CXOSNP Datasets in descending order of feature importance score. Features to the left and at the intersection of the red line are the optimum number of features chosen.

The presence of functional groups and electronegativity also significantly contribute to identifying the presence of intermolecular forces.

AdaBoost with Decision Tree

Figure 32 shows feature importance scores of AdaBoost(DT) where we select 13, 14, 21, and 27 for the C, CX, CXOS, and CXOSNP datasets. The feature selection in this model is similar to Random Forest with few extra additional features selected.

In the C dataset as shown in Figure 32a the highest feature importance score are for density, total bonds, and total rings. Similar to Random Forest the model chooses all features that describe the physical shape of the molecule using the Balaban J Index, eccentricity, sphericity, molecular weight, volume, and rotatable bonds. Carbon count has also been included in the selected features.

Also, triple bonds have a higher feature importance score than single bonds which was not expected since there are only 125 molecules with triple bonds greater than zero.

In the CX Dataset as shown in Figure 32b, the highest feature importance scores are for volume and total rings. It means the model finds a higher correlation between volume and total rings with melting points. It may be due to higher volume molecules having more rings and that may increase the melting point. Also, it may be due to halogen information being captured through the volume of the molecule since no atom count for halogen is selected.

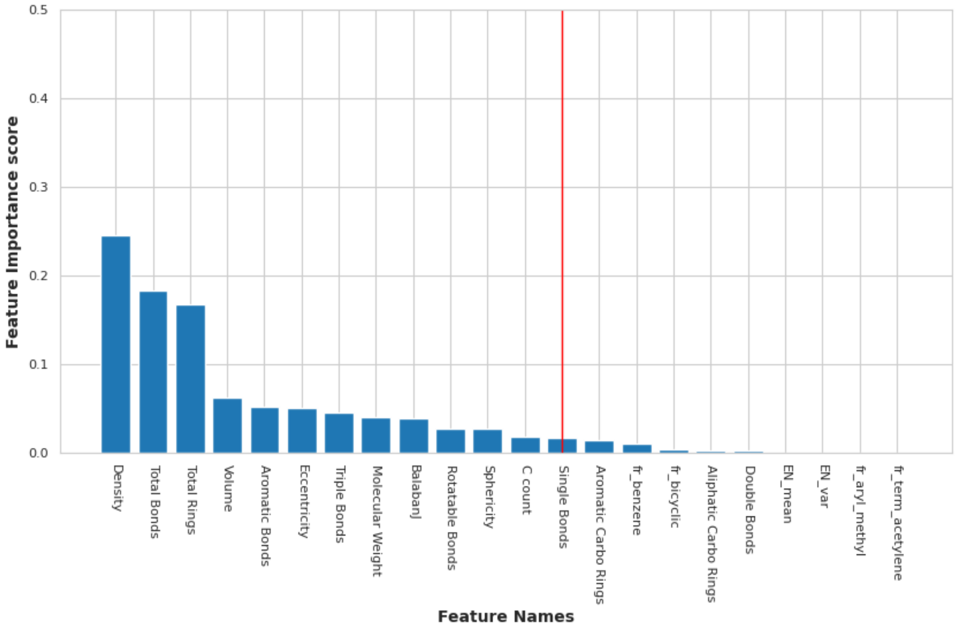
This model also chooses two electronegativity equations which are electronegativity difference mean and variance mean which may capture the polarity induced in the molecule.

In the CXOS dataset as shown in Figure 32c, Total Rings has the highest feature importance score. It is followed by density, total bonds, sphericity, single bonds, and aromatic bonds. It shows the model is capturing aromaticity using a combination of total rings, total bonds, and single and aromatic bonds.

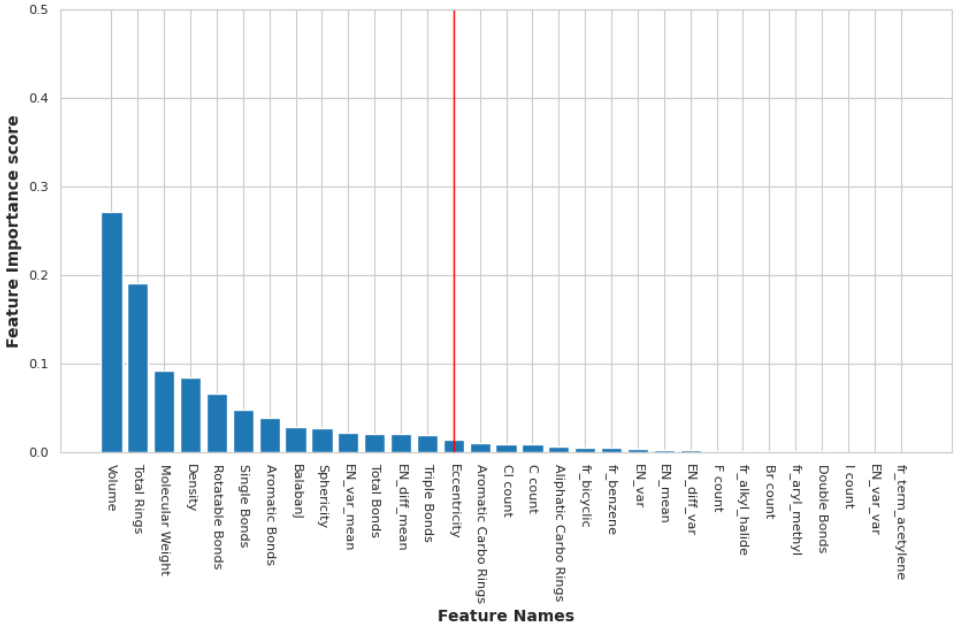
Hydrogen bonding features are also selected like ester, aromatic carboxyl, and aromatic hydroxyl groups. Since the model also selects aromatic carbon rings, it may be considering aromatic molecules with oxygen functional groups as important for the prediction of melting points.

In the CXOSNP dataset shown in Figure 32d the highest feature score is for total rings, followed by Total Bonds, and density. There are many fragments like amine, thiazole, bicyclic, benzene, ether, and aliphatic and aromatic carboxyl chosen which shows the model is dependent on predicting melting points by identifying the rings and the groups around it.

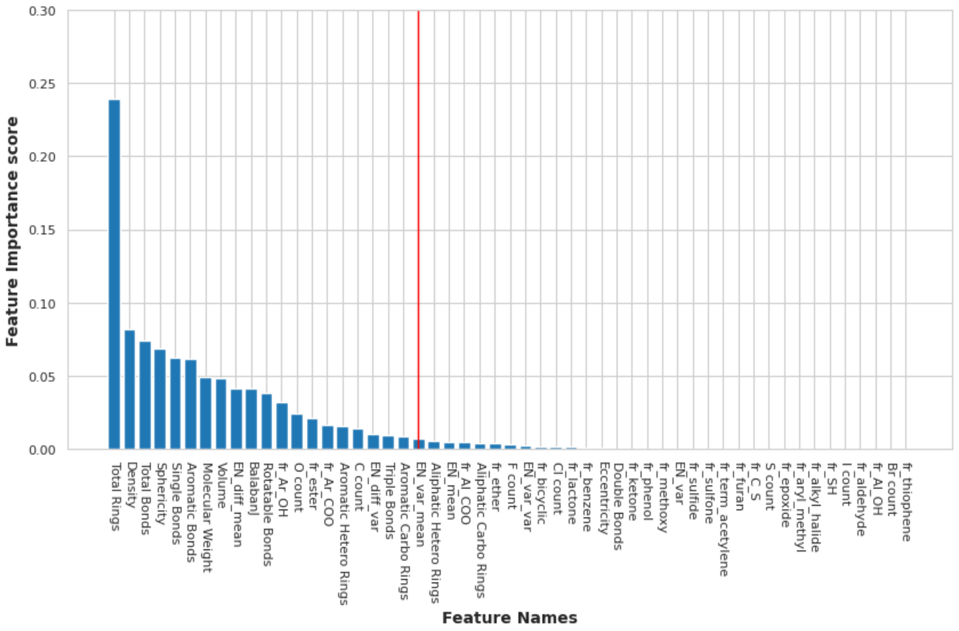
From the observation of feature analysis of AdaBoost(DT), we can assume the model



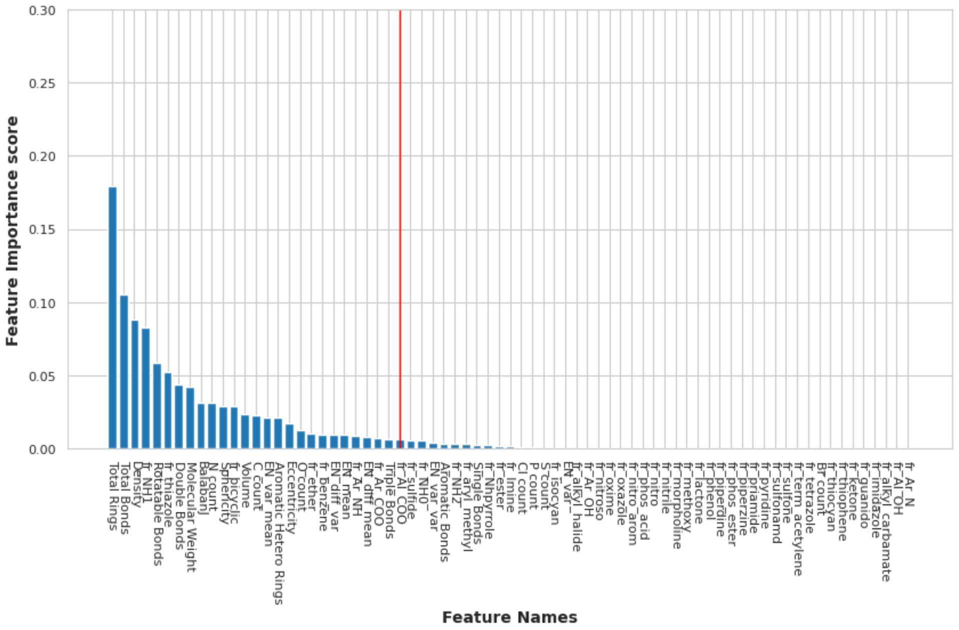
(a) Plot of Feature Importance Scores on C Dataset for AdaBoost(DT). 13 features are chosen.



(b) Plot of Feature Importance Scores on CX Dataset for AdaBoost(DT). 14 features are chosen.



(c) Plot of Feature Importance Scores on CXOS Dataset for AdaBoost(DT). 21 features are chosen.



(d) Plot of Feature Importance Scores on CXOSNP Dataset for AdaBoost (DT). Only the first 38 features are shown in descending order of feature score. 31 features are chosen.

Figure 32: Plots of Feature Importance scores for AdaBoost(DT) on C, CX, CXOS, and CXOSNP Datasets in descending order of feature importance score. Features to the left and at the intersection of the red line are the optimum number of features chosen.

considers physical shape, rings, and functional groups as features that have a correlation with predicting melting point.

XGBoost

XGBoost performance for all models has been good, even though they were overfitting. When we analyze the feature selection, this is the only model that chose the lowest number of features to reach the same level of test RMSE. Further, the features chosen are very distinct from Random Forest and AdaBoost(DT).

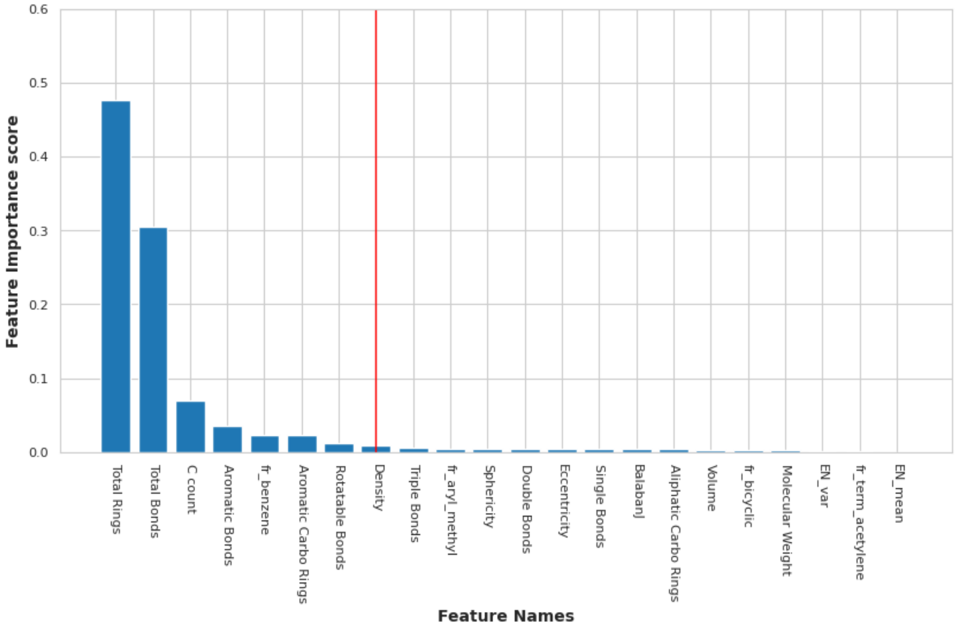
In the C dataset shown in Figure 33a, the highest importance scores are for total bonds and rings, which is similar to previous models. But this model does not choose a volume or molecular weight as important features since the score is lower for these features. Rather, it chose carbon count as the third important feature, followed by aromatic bonds, benzene fragments, and aromatic carbon rings. This shows the model is choosing features that can help distinguish aromatic rings and cyclic rings, and it is trying to identify the presence of benzene in the molecule, which may have a higher correlation with the prediction of melting point. It also chooses rotatable bonds and density.

In the CX dataset shown in Figure 33b, the highest importance is for volume. This is similar to previous models and confirms the fact that models can identify halogen atoms in the molecule using volume alone. Further, the inclusion of the electronegativity difference mean allows the model to identify the halogen much better and may indicate that polarity also has some correlation with the melting point.

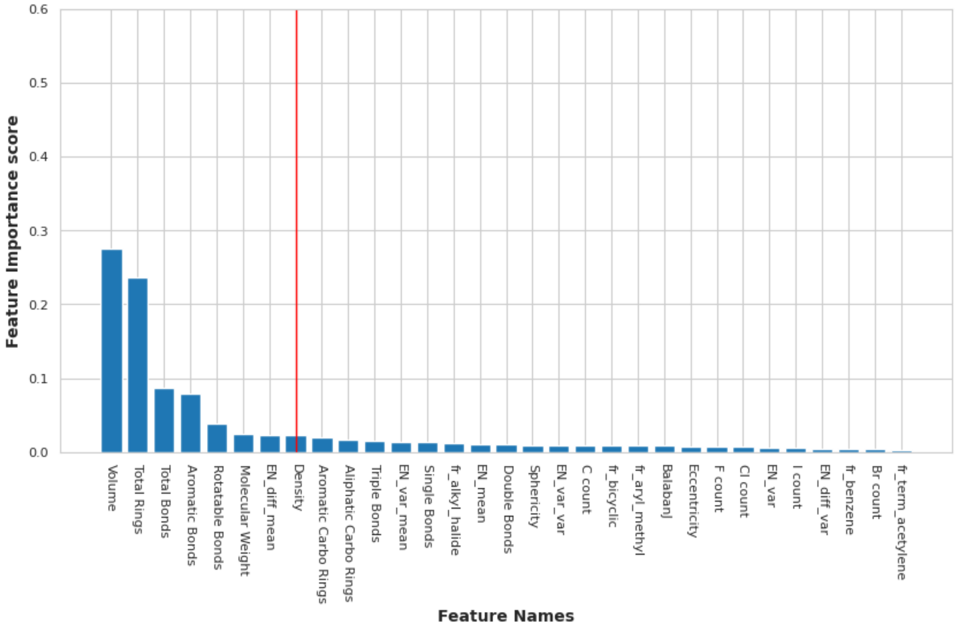
In the CXOS dataset shown in Figure 33c, the highest feature importance score is given to total bonds and total rings, which is similar to previous models. Beyond this, the XGBoost chooses substructures as more important features. We observe that functional groups like aromatic carboxyl, aliphatic and aromatic hydroxyl, ether, and ester are being chosen. Also, it chose phenol, furan, and lactone, which shows the model tries to predict melting points by identifying the presence of certain substructures in the molecule. Unlike previous models, the XGBoost does not give high importance to physical shape features.

In the CXOSNP dataset shown in Figure 33d, we can observe the trend of higher importance scores for substructures and functional groups compared to physical shape features. Since the model has higher nitrogen compounds, amine functional groups are also included in the feature selection. We observe that the substructures and functional groups selected are aliphatic and aromatic carboxyl, aliphatic and aromatic hydroxyl, ether, nitro, alkyl halide, isocyanate, phenol, nitrile, alkyl carbamate, ester, benzene, bicyclic, phosphoric acid, and piramide. It also chooses atom counts for nitrogen, oxygen, fluorine, and bromine.

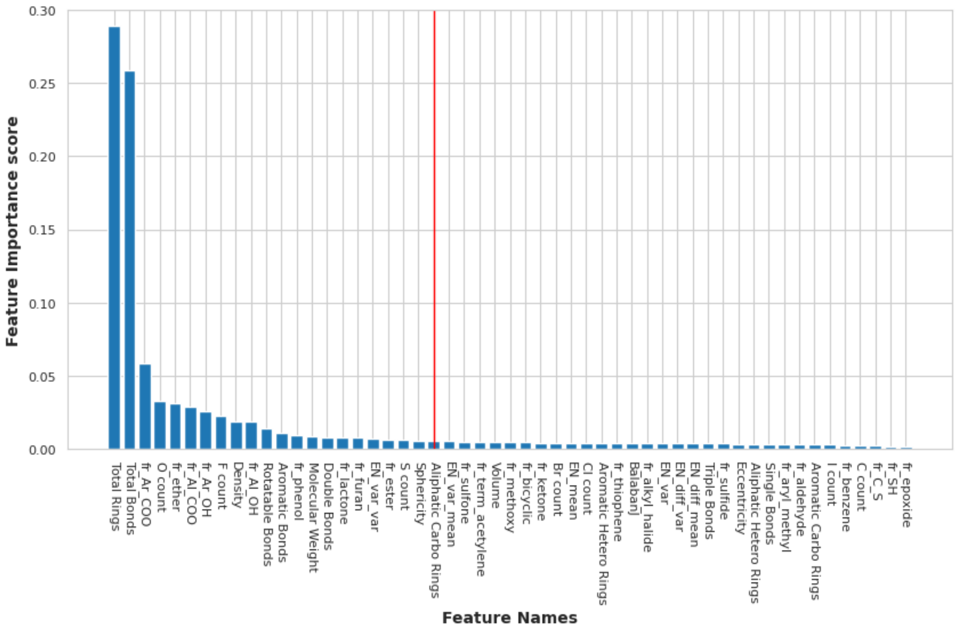
In XGBoost, the highest importance is given to total Rings and bonds. Among atom counts, it includes oxygen, fluorine, and sulfur. The model gives more importance to many functional groups like ether, aliphatic carboxyl, aromatic hydroxyl, phenol, lactone,



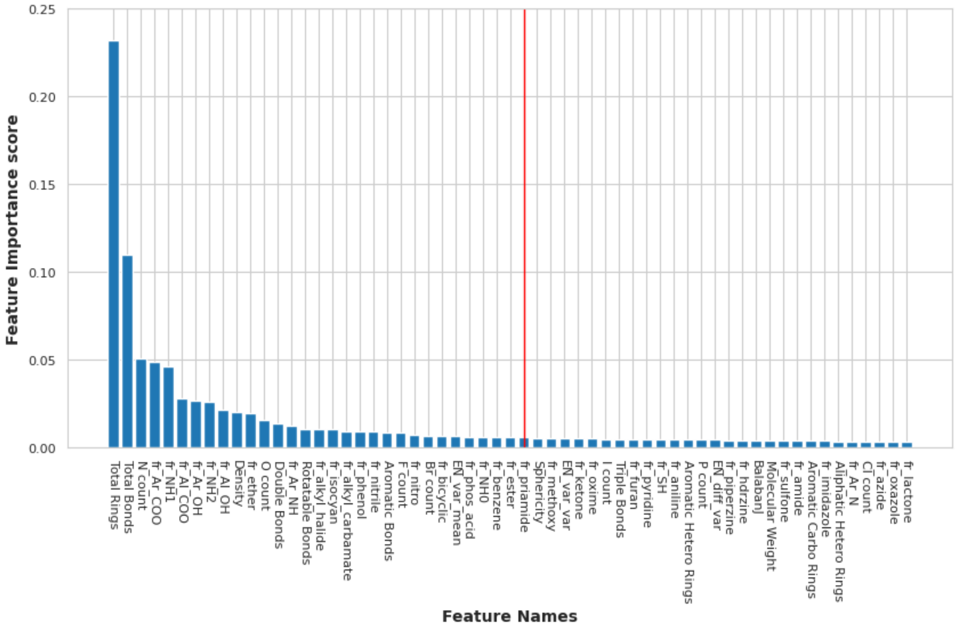
(a) Plot of Feature Importance Scores on C Dataset for XGBoost. 8 features are chosen.



(b) Plot of Feature Importance Scores on CX Dataset for XGBoost. 8 features are chosen.



(c) Plot of Feature Importance Scores on CXOS Dataset for XGBoost. 22 features are chosen.



(d) Plot of Feature Importance Scores on CXOSNP Dataset for XGBoost. Only the first 38 features are shown in descending order of feature score. 31 features are chosen.

Figure 33: Plots of Feature Importance scores for XGBoost on C, CX, CXOS, and CXOSNP Datasets in descending order of feature importance score. Features to the left and at the intersection of the red line are the optimum number of features chosen.

furan, and ester. It is observed that XGBoost prioritizes functional group features more than others.

Observations for XGBoost on all these datasets show this model learns better from the substructural group's presence than the physical shape description of the molecule. This proves that melting points can also be identified by describing the substructures, which are very similar to Morgan Fingerprint and Mol2Vec's models. Also, the XGBoost model can accurately perform with the least features compared to all other models making it faster than others due to lesser data size requirements.

Principal Component Analysis

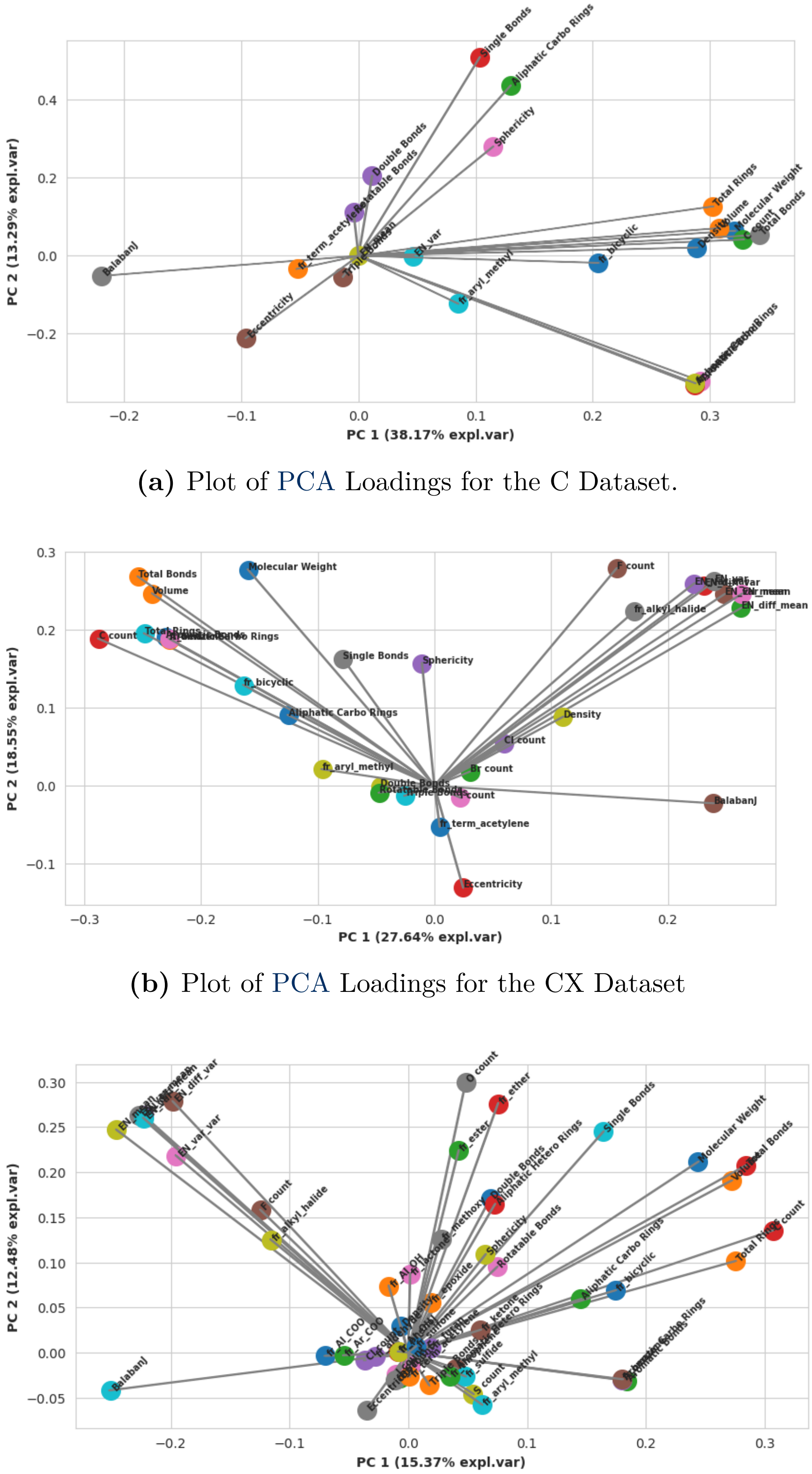
SVR, AdaBoost(SVR) and ANN models for descriptive features use PCA for feature selection.

Figure 34a shows the PCA loadings plot for the C dataset. We observe total rings, density, volume, molecular weight, carbon count and total bonds are highly correlated to each other. Total Bonds have the strongest correlation. Even the bicyclic substructure is in the same direction as others which may be due to the bicyclic structure having many bonds that increase the total number of bonds in the molecule. Balaban J Index is highly negatively correlated to total bonds and others which shows the component's relation cannot be explained by those features. This can be explained by the fact that Balaban J Index is a graphical descriptor and is not related to bonds and rings.

Eccentricity and sphericity are on opposite ends to each other which is accurate since spherical shapes will have sphericity as 1 and eccentricity as 0. It is also observed that sphericity is in the same direction as single bonds and aliphatic carbon rings which shows it has some positive correlation with these features. This may indicate that there are molecules that may be in the cage structure or maybe the shape due to branching being closer to spherical. Similarly, aromatic bonds and aromatic carbon rings are positively correlated with each other.

Figure 34b shows PCA loadings for the CX Dataset. We observe that fluorine and alkyl halides have a positive correlation with electronegativity features, which is related to the fact that fluorine is the most electronegative atom and its presence will impact the electronegativity equations. Further, we observe that total rings and aromatic carbon rings have a strong positive correlation, which may indicate there are many aromatic ring molecules in the dataset.

In the PCA loadings plot for the CXOS dataset shown in Figure 34c, apart from the previous relationships, we observe a strong correlation between oxygen and its functional groups like ether and ester. Since there are more oxygen compounds, it may influence the principal components. All other fragments are clustered around the center and do not have strong correlations.



(a) Plot of PCA Loadings for the C Dataset.

(b) Plot of PCA Loadings for the CX Dataset

(c) Plot of PCA Loadings for the CXOS Dataset.

Figure 34: PCA Loading Plot for the C, CX, and CXOS Datasets.

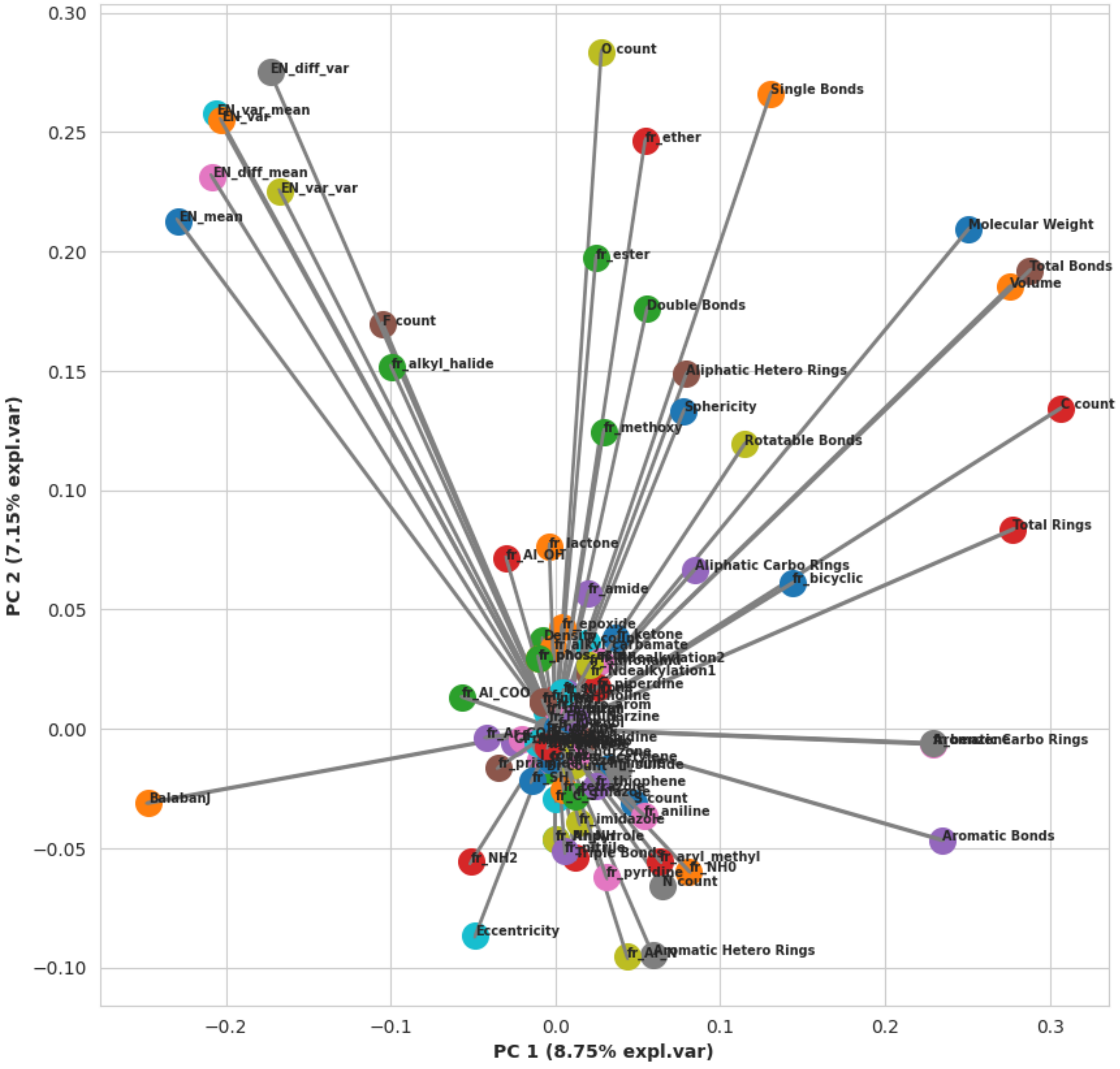


Figure 35: PCA Loading Plot for the CXOSNP Dataset.

We see similar results in Figure 35 which shows PCA loadings plot for the CXOSNP dataset. There is a very weak correlation for the fragments except for oxygen functional groups like ether, ester, and methoxy groups. There is a strong correlation between nitrogen count and its functional group and is in the same direction as aromatic hetero rings. This may indicate that there are many nitrogen molecules that may be part of hetero aromatic rings and can explain the variance in PCA components.

There is a correlation between the secondary amine group and eccentricity which may indicate the shape of the molecule being flatter for molecules with the presence of a secondary amine group.

From the observations of the PCA loadings plot, we can conclude that the first few PCA components are explained by molecular weight, total volume, total rings, carbon count, electronegativity, oxygen count, aromaticity, and other physical shape descriptive features. Also, functional groups and fragments for oxygen and nitrogen compounds explain the other PCA components. This indicates for a dataset that comprises a specific heteroatom in more molecules will have a stronger correlation in assisting the prediction of melting points. The other fragments have a weak correlation and therefore do not explain the variance in the dataset.

7.2 Future Improvements

Our models demonstrated good performance on the datasets; however, we noticed that the feature importance scores for electronegativity features were not high. This suggests that these features may not contribute significantly to the prediction of melting point. Therefore, our attempt to capture polarity induced in the molecule and detect intermolecular interactions based on differences in electronegativity may not be effective.

To address this, we explored the use of Gasteiger charges, which calculate partial charges assigned to individual atoms in a molecule based on electronegativity and electron distribution [50]. By employing RDKit, we found that the charges assigned to each atom provided a better representation of polarity induced in the molecule compared to our electronegativity features. However, unlike models that utilize the charges as an array representation for each atom [16]), we encountered a challenge in incorporating Gasteiger charges due to the need to calculate the mean or variance of partial charges. This approach might diminish the quality of information as it fails to capture the positional information of atom charges, such as distinguishing between edge and central atom charges. The precise position of an atom's charge is crucial in identifying intermolecular interactions between molecules. As a result, further analysis is required to effectively incorporate Gasteiger charges into our feature set.

In our feature set, we encountered difficulty in capturing information about stereoisomerism. Stereoisomerism refers to compounds that share the same molecular formula and connectivity of atoms but differ in the spatial arrangement of atoms or groups, leading to distinct melting points. Although stereoisomerism can be represented by the coordinate positional information of atoms in a distance matrix, incorporating this information would complicate our goal of using simple numerical features. Some models have addressed stereoisomerism by encoding the chirality of atoms in an array representation [16], enabling the identification of molecules with stereoisomers.

We explored various methods available in RDKit; however, solving this problem using our numerical feature-based approach proved challenging. It would require comparing each stereoisomer to all other molecules in the dataset, making it impractical to represent stereoisomerism as a single feature denoting true or false. One possible solution is to employ unsupervised methods, such as clustering, to screen the dataset and identify molecules that are stereoisomers of each other. Subsequently, similarity index numbers could be assigned to these molecules, facilitating the capture of stereoisomerism. However, implementing this method would necessitate in-depth analysis and constitute a separate research problem to be addressed.

Our approach in identifying hydrogen bonding by counting the presence of functional groups has proven partially helpful but does not fully address the broader problem. One limitation of our approach is that we capture the presence of hydrogen bonding but not its location. The strength of hydrogen bonding can be influenced by factors such as the location of electronegative atoms and the proximity of hydrogen atoms to other molecules. To properly analyze hydrogen bonding strength, it is necessary to study molecular crystals rather than individual molecules. Unfortunately, due to the limitations of working with molecules in SMILES format, we were unable to convert them into their crystal structures. As a result, our model can only capture the presence of hydrogen bonding but cannot explain its strength accurately. However, we did observe that our models selected amine, carboxyl, and hydroxyl groups and found a correlation between these groups and the melting point temperature.

To study hydrogen bonding and its strength more effectively, it is advisable to analyze molecular crystals, which provide distance information between two molecules. This distance information can be used to compute the distances between an electronegative atom of one molecule and a hydrogen atom of another molecule within their proximity, enabling the determination of hydrogen bonding strength and assigning it an index number. Additionally, it would be possible to rank the hydrogen bonding strength according to the atom or functional group in the molecule. However, this approach requires a dataset comprising molecular crystals and necessitates a comprehensive dataset, combining the ONS and CSD datasets, to achieve the desired richness of data.

Addressing these challenges will significantly enhance our ability to predict melting points with greater confidence. We should also explore alternative formats that can capture the relevant characteristics of molecules more effectively. As we have observed, algorithms like Morgan Fingerprint and Mol2Vec have demonstrated good accuracy in predicting melting points by effectively identifying substructural patterns. Therefore, we can draw inspiration from these approaches and explore new methods that go beyond numerical features to capture the intricate details of molecules and their interactions.

8 Conclusion

The final goal of the research was to perform feature engineering and create a generalized model for the prediction of melting points for unseen organic molecules. We create segregated datasets: C, CX, CXOS, and CXOSNP datasets from a combination of CSD and ONS datasets by only selecting organic molecules. For each molecule, we create features to describe its shape, size, electronegativity, flexibility, and substructure patterns to identify intermolecular interactions.

We trained Random Forests, AdaBoost with base estimators as DT and SVR, XGBoost and ANN models on all four datasets. We observe that Random Forest and XGBoost have the lowest test RMSE. These models have the best performance on the test set, but they also show evidence of overfitting the training data. SVR and ANN are the most generalized models with the least difference between train and test R^2 scores and RMSE values, but their RMSE values are higher than Random Forest and XGBoost.

For the CXOSNP dataset, we achieved a test RMSE of 42.63 ± 2.36 K for XGBoost which was able to outperform the Morgan Fingerprint and Mol2vec algorithm with test RMSE of 44.96 ± 10.53 K and 45.56 ± 1.47 K, respectively. Also, we were able to achieve these values for XGBoost with a lower feature size of 31 features per molecule, compared to Morgan Fingerprint and Mol2Vec algorithms that require 256 and 300 array sizes per molecule, respectively. This shows that our features have selective information that helps in achieving data compression, thus reducing the amount of storage data required per molecule.

Further, during model analysis using feature selection, we identified that Random Forest and XGBoost have the highest feature importance scores for total rings. Random Forest learns about physical shape features like total bonds, density, molecular weight, volume, sphericity, and eccentricity to predict the melting point temperature. It gives lower scores for substructures, therefore does not depend on intermolecular interaction features. On the contrary, XGBoost learns more from substructures and gives lower importance to physical shape features. It finds a higher correlation with melting point temperature by identifying the presence of a substructure in the molecule that may indicate it learns more from intermolecular interaction features.

From the analysis of PCA, we were able to understand the relationships between the features created. We identified that total rings, total bonds, molecular weight, and volume have a positive correlation with each other and explain the variance in the first few components. Electronegativity equations and fluorine atom count also have a positive correlation with each other, showing that fluorine's electronegativity is the highest. Since the dataset has a large number of molecules with oxygen and nitrogen, we observed a positive correlation between oxygen and nitrogen with their functional groups, where oxygen has a stronger relationship with ether and ester groups and nitrogen with amine groups.

From these observations, we can conclude that the melting point of a molecule can be determined either by describing the shape of the molecule with more refinement for a model trained on Random Forest, or by identifying all substructural patterns in the molecule for a model trained on XGBoost. Although we had limitations in identifying stereoisomerism and polarity with confidence, our features, and models performed well and achieved a low RMSE value of 42.63 K for a large subset of organic molecules with highly complex structures and a large variance in melting points.

References

[1] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al., "Pubchem 2023 update," *Nucleic Acids Research*, vol. 51, no. D1, pp. D1373–D1380, 2023.

[2] Alexandre Varnek and Igor Baskin, *Fragment Descriptors in SAR/QSAR/QSPR Studies, Molecular Similarity Analysis and in Virtual Screening*, 09 2008.

[3] Wikipedia contributors, "Support vector machine," April 2023, Accessed on : April 26,2023.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] Sundarraj Vijayaraghavan, "Random forest regression," https://levelup.gitconnected.com/random-forest-regression-209c0f354c84, 2019, Accessed on: April 28, 2023.

[6] Will Koehrsen, "The ultimate guide to adaboost, random forests, and xgboost," https://towardsdatascience.com/the-ultimate-guide-to-adaboost-random-forests-and-xgboost-7f9327061c4f, 2020, Accessed on: April 28, 2023.

[7] "Artificial neural networks and its applications," https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/, Accessed on : May 13, 2023.

[8] Katherine A Chu and Samuel H Yalkowsky, "An interesting relationship between drug absorption and melting point," *International journal of pharmaceutics*, vol. 373, no. 1-2, pp. 24–40, 2009.

[9] Edward Harvel Kerns, Li Di, and Edward H Kerns, *Drug-like properties: concepts, structure design and methods*, vol. 10, Academic press New York, 2008.

[10] Neera Jain and Samuel H Yalkowsky, "Estimation of the aqueous solubility i: application to organic nonelectrolytes," *Journal of pharmaceutical sciences*, vol. 90, no. 2, pp. 234–252, 2001.

[11] Stew Dent, "Purity and identification of solids using melting points," *Department of Chemistry Portland State University Portland*, 2006.

[12] Nafisur Rahman, Syed Najmul Hejaz Azmi, and Hui-Fen Wu, "The importance of impurity analysis in pharmaceutical products: an integrated approach," *Accreditation and Quality Assurance*, vol. 11, pp. 69–74, 2006.

[13] A. D. McNaught and A. Wilkinson, "Compendium of chemical terminology," *The Gold Book*, vol. 2, no. 66, 1997.

[14] Samuel H Yalkowsky, "Carnelley's rule and the prediction of melting point," *Journal of pharmaceutical sciences*, vol. 103, no. 9, pp. 2629–2634, 2014.

[15] Robert Abramowitz and Samuel H Yalkowsky, "Melting point, boiling point, and symmetry," *Pharmaceutical research*, vol. 7, pp. 942–947, 1990.

[16] Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen, "Convolutional embedding of attributed molecular graphs for physical property prediction," *Journal of chemical information and modeling*, vol. 57, no. 8, pp. 1757–1772, 2017.

[17] Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong, "Graph networks as a universal machine learning framework for molecules and crystals," *Chemistry of Materials*, vol. 31, no. 9, pp. 3564–3572, 2019.

[18] Laura D Hughes, David S Palmer, Florian Nigsch, and John BO Mitchell, "Why are some properties more difficult to predict than others? a study of qspr models of solubility, melting point, and log p," *Journal of chemical information and modeling*, vol. 48, no. 1, pp. 220–232, 2008.

[19] Colin R Groom, Ian J Bruno, Matthew P Lightfoot, and Suzanna C Ward, "The cambridge structural database," *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, vol. 72, no. 2, pp. 171–179, 2016.

[20] Sydney R Hall, Frank H Allen, and I David Brown, "The crystallographic information file (cif): a new standard archive file for crystallography," *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 47, no. 6, pp. 655–685, 1991.

[21] Jean-Claude Bradley, Andrew Lang, Antony Williams, and Evan Curtin, "Ons open melting point collection," *Nature Precedings*, pp. 1–1, 2011.

[22] David Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[23] Steven H Bertz, "The first general index of molecular complexity," *Journal of the American Chemical Society*, vol. 103, no. 12, pp. 3599–3601, 1981.

[24] Greg Landrum, "Rdkit documentation," *Release*, vol. 1, no. 1-79, pp. 4, 2013.

[25] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison, "Open babel: An open chemical toolbox," *Journal of cheminformatics*, vol. 3, no. 1, pp. 1–14, 2011.

[26] Carol Higginbotham, "Melting and boiling points," Open Oregon Educational Resources, Online; Accessed: 2023-04-28.

[27] RH Valentine, GE Brodale, and WF Giauque, "Trifluoromethane: entropy, low temperature heat capacity, heats of fusion and vaporization, and vapor pressure1," *The Journal of Physical Chemistry*, vol. 66, no. 3, pp. 392–395, 1962.

[28] R Todeschini, V Consonni, and J Gasteiger, "Handbook of chemoinformatics: from data to knowledge in 4 volumes," 2003.

[29] RDKit Contributors, "Rdkit: Chem.fragments module," http://rdkit.org/docs/source/rdkit.Chem.Fragments.html, 2021, Accessed on: April 8, 2023.

[30] Alexandru T Balaban, "Topological indices based on topological distances in molecular graphs," *Pure and applied chemistry*, vol. 55, no. 2, pp. 199–206, 1983.

[31] RDKit Contributors, "RDKit.Chem.Descriptors3D module," http://rdkit.org/docs/source/rdkit.Chem.Descriptors3D.html, 2023, Accessed on: 8-April-2023.

[32] Martin Knor, R Škrekovski, and Aleksandra Tepeh, "Mathematical aspects of balaban index," *MATCH Commun. Math. Comput. Chem*, vol. 79, pp. 685–716, 2018.

[33] David Rogers and Mathew Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.

[34] Harry L Morgan, "The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service.," *Journal of chemical documentation*, vol. 5, no. 2, pp. 107–113, 1965.

[35] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas, "Molecular fingerprint similarity search in virtual screening," *Methods*, vol. 71, pp. 58–63, 2015.

[36] Seyed Aghil Hooshmand, Sadegh Azimzadeh Jamalkandi, Seyed Mehdi Alavi, and Ali Masoudi-Nejad, "Distinguishing drug/non-drug-like small molecules in drug discovery using deep belief network," *Molecular Diversity*, vol. 25, pp. 827–838, 2021.

[37] Isidro Cortes-Ciriano, "Bioalerts: a python library for the derivation of structural alerts from bioactivity and toxicity data sets," *Journal of cheminformatics*, vol. 8, no. 1, pp. 1–6, 2016.

[38] Hyun Woo Kim, Mingxun Wang, Christopher A Leber, Louis-Félix Nothias, Raphael Reher, Kyo Bin Kang, Justin JJ Van Der Hooft, Pieter C Dorrestein, William H Gerwick, and Garrison W Cottrell, "Npclassifier: A deep neural network-based structural classification tool for natural products," *Journal of Natural Products*, vol. 84, no. 11, pp. 2795–2807, 2021.

[39] Sabrina Jaeger, Simone Fulle, and Samo Turk, "Mol2vec: unsupervised machine learning approach with chemical intuition," *Journal of chemical information and modeling*, vol. 58, no. 1, pp. 27–35, 2018.

[40] John J Irwin and Brian K Shoichet, "Zinc- a free database of commercially available compounds for virtual screening," *Journal of chemical information and modeling*, vol. 45, no. 1, pp. 177–182, 2005.

[41] Anna Gaulton, Anne Hersey, Michał Nowotka, A Patricia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J Bellis, Elena Cibrián-Uhalte, et al., "The chembl database in 2017," *Nucleic acids research*, vol. 45, no. D1, pp. D945–D954, 2017.

[42] Tommaso Galeazzo and Manabu Shiraiwa, "Predicting glass transition temperature and melting point of organic compounds via machine learning and molecular embeddings," *Environmental Science: Atmospheres*, vol. 2, no. 3, pp. 362–374, 2022.

[43] William S Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[44] Sebastian Raschka, *Python machine learning*, Packt publishing ltd, 2015.

[45] Rameswar Debnath and Haruhisa Takahashi, "Kernel selection for the support vector machine," *IEICE transactions on information and systems*, vol. 87, no. 12, pp. 2903–2904, 2004.

[46] Leo Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996.

[47] Yoav Freund and Robert E Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[48] Robert E Schapire, "Explaining adaboost," *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pp. 37–52, 2013.

[49] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al., "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[50] Johann Gasteiger and Mario Marsili, "Iterative partial equalization of orbital electronegativity—a rapid access to atomic charges," *Tetrahedron*, vol. 36, no. 22, pp. 3219–3228, 1980.

The page is intentionally left blank.

A Scatter Plots

Scatter plots for the C, CX, CXOS, and CXOSNP datasets.

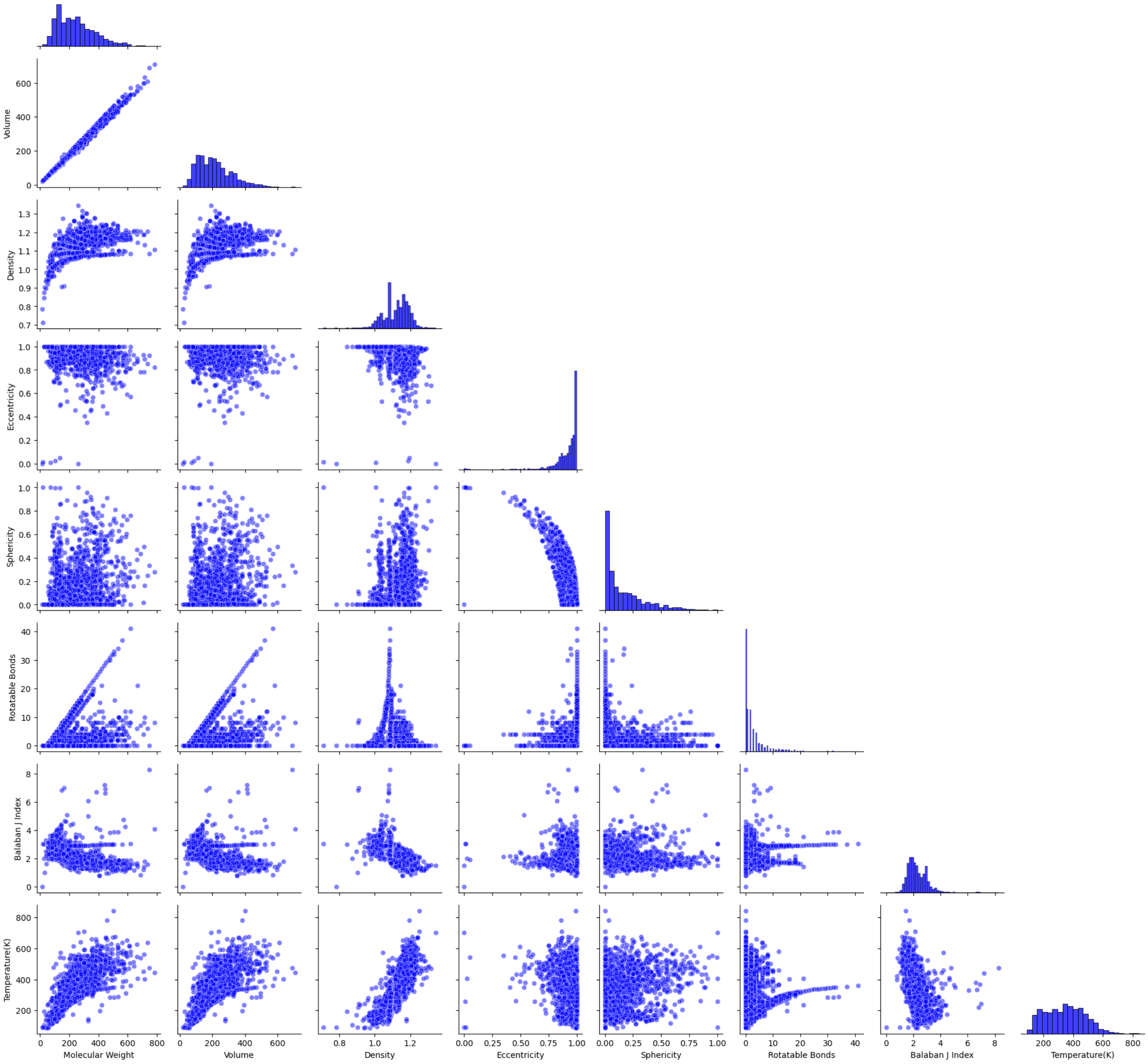


Figure 36: Scatter plot of C Dataset.

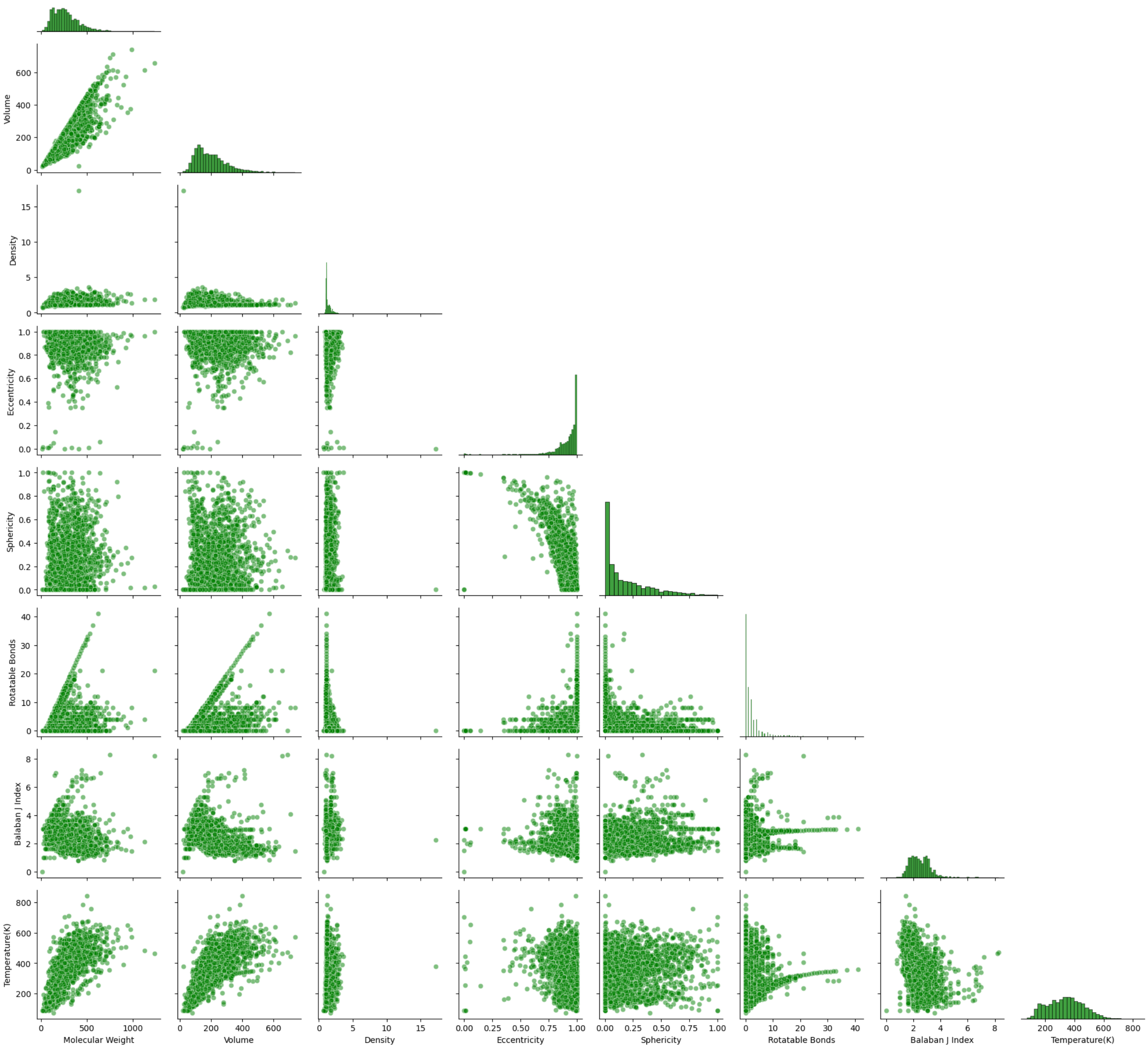


Figure 37: Scatter plot of CX Dataset.

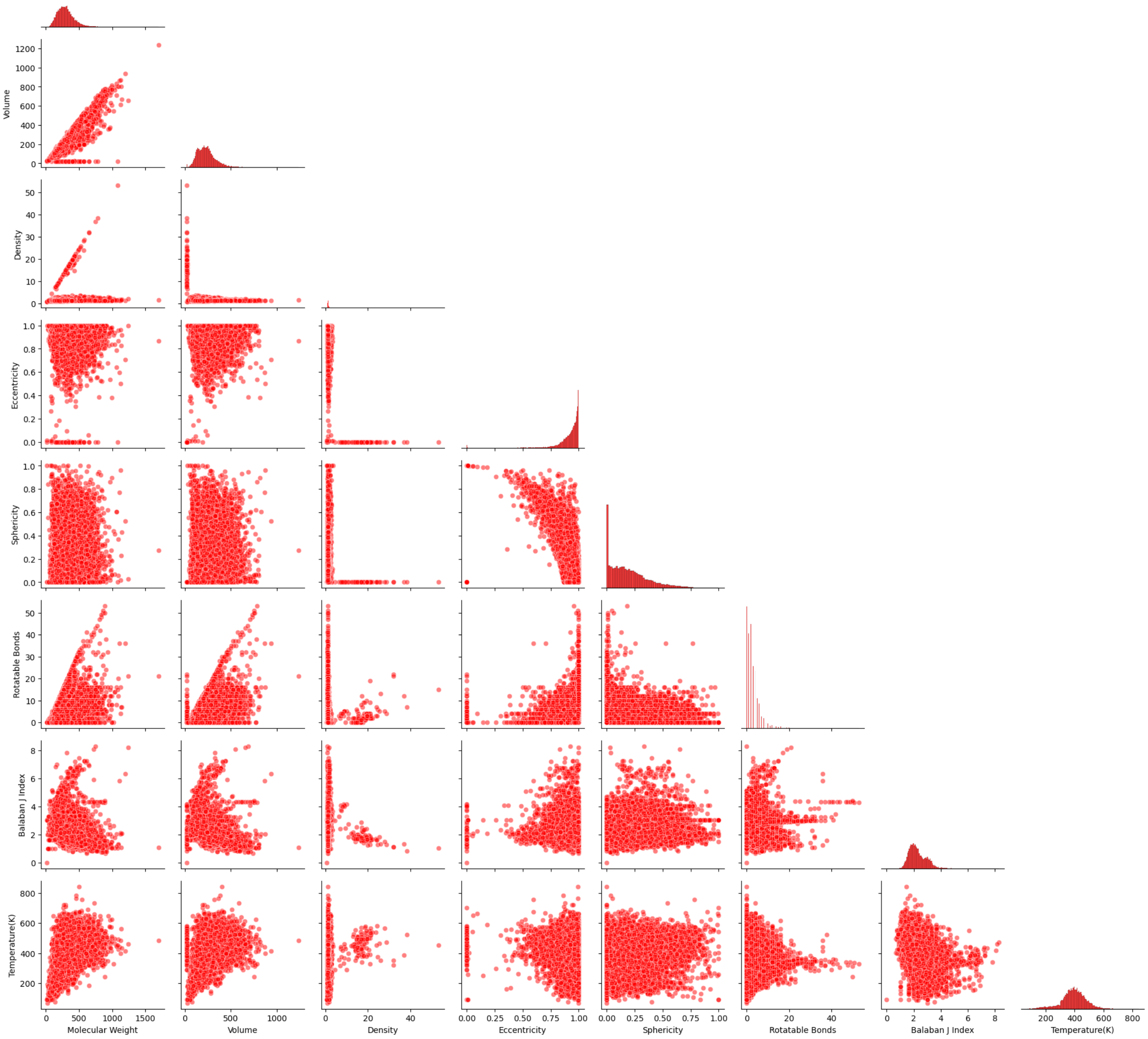


Figure 38: Scatter plot of CXOS Dataset.

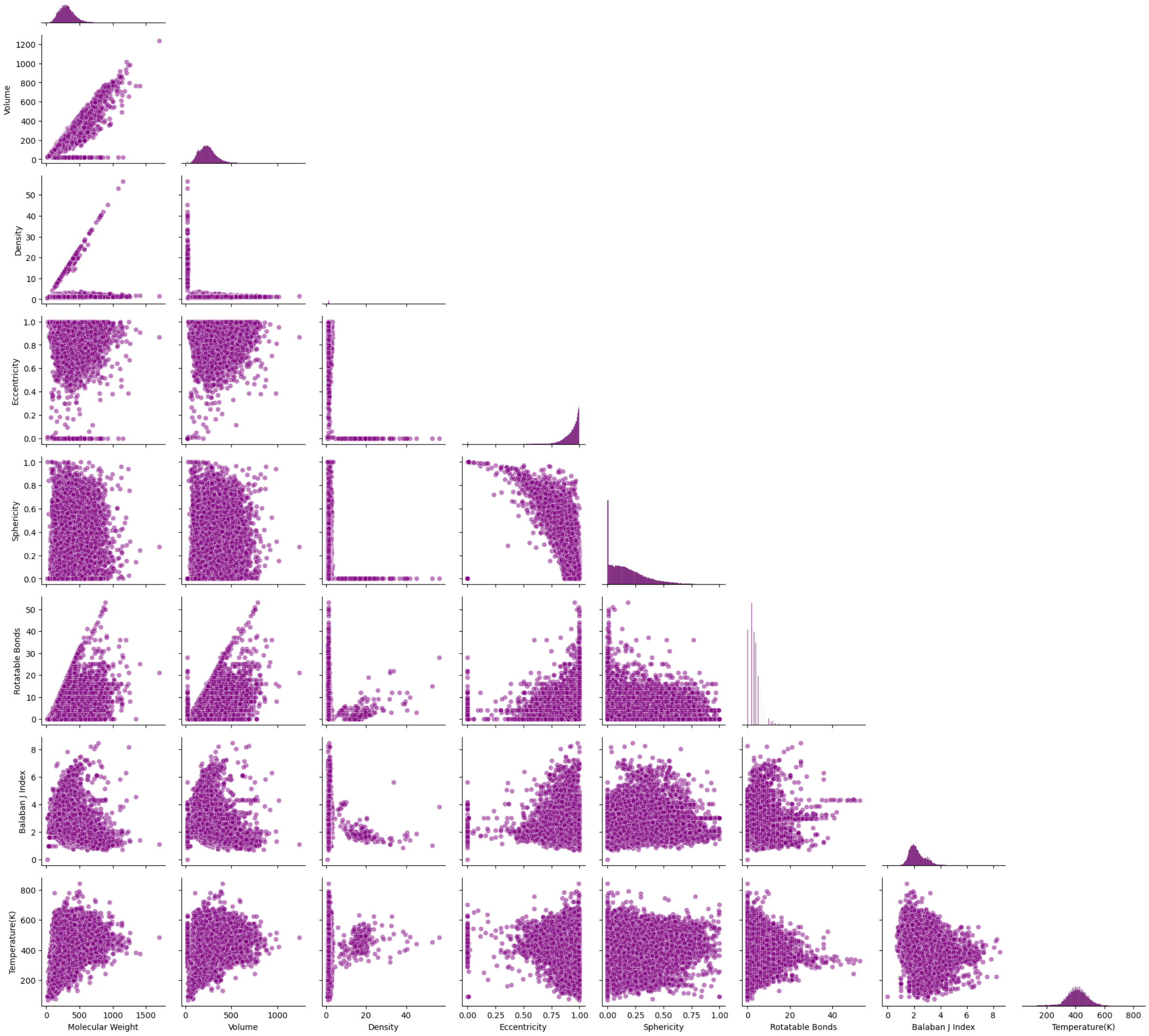
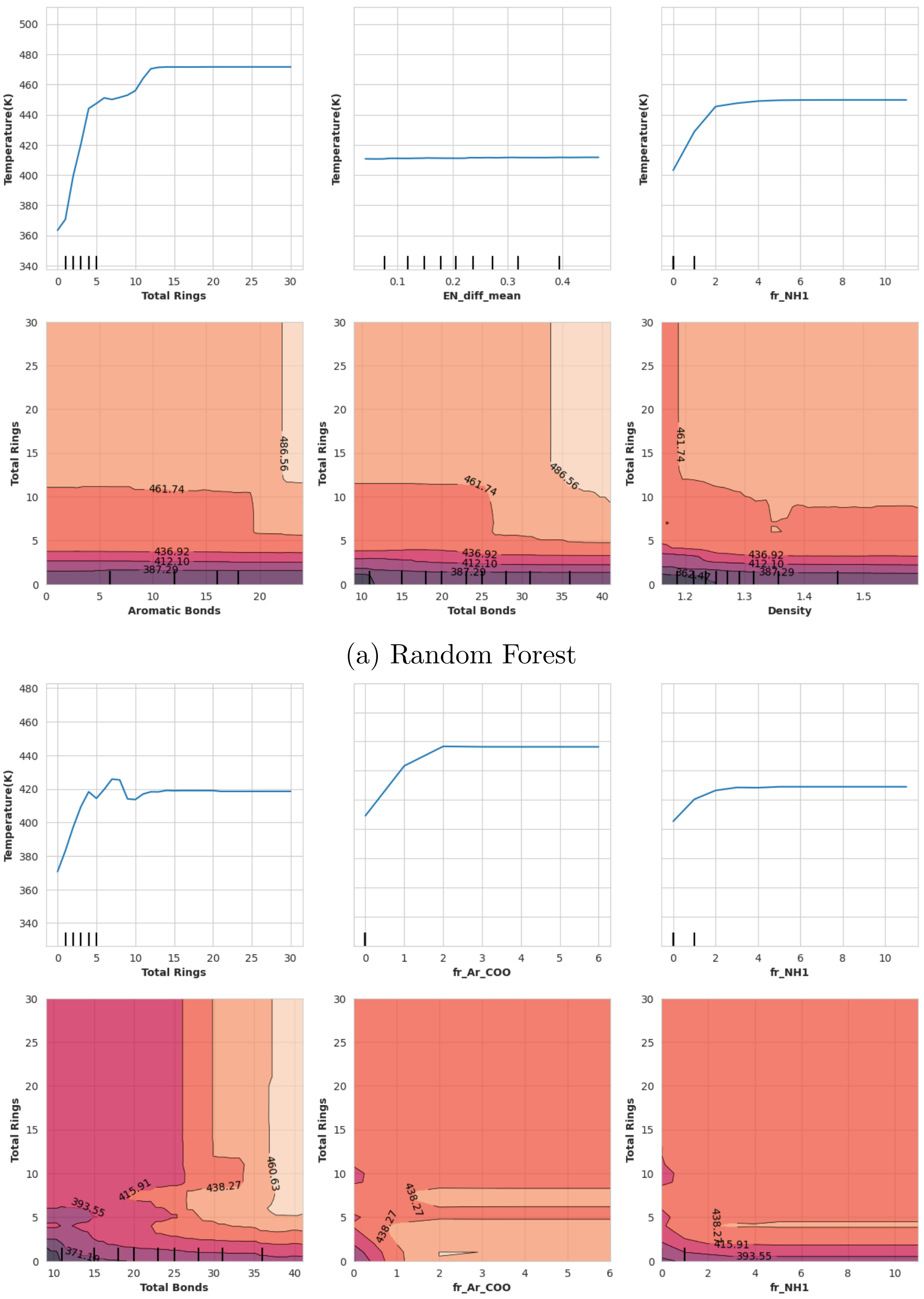


Figure 39: Scatter plot of CXOSNP Dataset.

B Partial Dependence Plot

Partial Dependence Plot for XGBoost and Random Forest on the CXOSNP Dataset that



(a) Random Forest

(b) XGBoost

Figure 40: Partial Dependence Plot of Random Forest and XGBoost on the CXOSNP Dataset.

The page is intentionally left blank.

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway