Viscosity Prediction of Organic Liquids Using Machine Learning Models

submitted

by

Jawaji Srikanth

(210107036)

under the

guidance of

Prof. Pallab Ghosh

Department of Chemical Engineering

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI,

GUWAHATI - 781039, ASSAM

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
DEPARTMENT OF CHEMICAL ENGINEERING
GUWAHATI, 781039, ASSAM, INDIA

Date: 22/11/2024

CERTIFICATE

It is certified that the work contained in the thesis entitled "Viscosity Prediction of Organic Liquids Using Machine Learning Models", is a Bonafide work of **Mr. Jawaji Srikanth (Roll No: - 210107036)** and has been carried out in the Department of Chemical Engineering, Indian Institute of Technology Guwahati, under my supervision and this work has not been submitted elsewhere for any other degree.

Prof. Pallab Ghosh
Department of Chemical Engineering
Indian Institute of Technology Guwahati

Acknowledgement

I would like to express my regards and gratitude to my thesis supervisor, **Prof. Pallab Ghosh**, Department of Chemical Engineering, Indian Institute of Technology Guwahati, Guwahati. He has been a constant source of well-needed guidance, suggestive discussions and helpful advice.

I want to thank the Department of Chemical Engineering, IIT Guwahati, for their support.

Jawaji Srikanth
(210107036)

Contents

Abstract

Viscosity is a fundamental property of organic liquids, and is important for a wide range of industrial applications like chemical manufacturing, petrochemical processing, pharmaceuticals and food production. Accurate determination of viscosity is very important for optimizing processes like mixing, heat transfer and material flow, which directly influence product quality and operational efficiency in industries. While many empirical and theoretical correlations exist for predicting viscosity, they are often limited in scope, and rely on simplified assumptions that may fail to capture the complex, nonlinear relationships between molecular structures, nature of molecules and many other influencing factors. Also, these correlations may lack generalizability across diverse chemical systems such as polar and non-polar liquids or liquids with complex molecular structures and might require extensive experimental data for parameter fitting.

Machine Learning (ML) techniques offer us a powerful alternative by learning from experimental data that is already available to develop models capable of handling the complex dependencies and variations inherent in various types of organic liquids. ML-based models can quickly and accurately predict viscosity across a broad spectrum of conditions, which also helps reducing the reliance on experimental methods and outdated correlations. These models also provide the flexibility, scalability and the ability to adapt to novel compounds or compositions that lack prior empirical correlations. This study mainly focuses on the development and validation of ML models for predicting the viscosity of organic liquids, showcasing their ability to enhance industrial decision-making, streamline processes and accelerate innovation. By integrating ML techniques, industries can overcome the limitations of traditional approaches, and also achieving higher accuracy, efficiency and cost-effectiveness in viscosity estimation.

Introduction

Viscosity, often referred to as a fluid's "thickness," it is a measure of its resistance to flow when the liquid is subjected to an external force. Fundamentally, viscosity quantifies the internal friction within a fluid. When a shearing stress is applied to a fluid, its layers move at different velocities creating a velocity gradient. The ratio of the applied shear stress to this velocity gradient defines the viscosity. In simpler words, fluids with lower viscosity, like water, flow easily, while those with higher viscosity, like honey or motor oil, resist motion more significantly.

On a molecular level, viscosity arises from the interactions and collisions between molecules in a fluid. Increased viscosity means that each layer of the fluid exerts a stronger frictional drag on adjacent layers, reducing the velocity gradient. Unlike static properties such as density, viscosity is a nonequilibrium property, meaning it can only be measured when the fluid is in motion. Despite this, viscosity, much like density, reflects the fluid's thermodynamic state and is influenced by factors such as temperature, pressure, and composition.

Industrial Relevance

Accurate knowledge of viscosity is critical across multiple industries due to its direct impact on process efficiency, product quality, and equipment performance:

Chemical Manufacturing: In reactors, viscosity influences mixing efficiency and heat transfer. For instance, highly viscous reactants require specific stirring mechanisms to ensure uniform temperature distribution and prevent localized overheating.

Petrochemical Industry: The viscosity of fuels and lubricants determines their flow characteristics under varying temperatures. For example, engine oils must maintain optimal viscosity at both high operating temperatures and low startup temperatures to ensure proper lubrication and minimize wear.

Food and Beverage Industry: Viscosity dictates the texture, stability, and sensory appeal of products like sauces, syrups, and dairy items. A consistent viscosity ensures uniform mixing and packaging processes, as well as consumer satisfaction.

Pharmaceuticals: The flow properties of liquid drugs and syrups depend heavily on viscosity. Precise control ensures the correct dosage and delivery during manufacturing.

Paints and Coatings: The application properties of paints, such as how smoothly they spread and adhere, are governed by viscosity. Incorrect viscosity can lead to uneven application or prolonged drying times.

Why Accurate Viscosity Measurements Matter

Inaccuracies in viscosity data can lead to inefficiencies and financial losses. For example, overestimating the viscosity of a fluid can result in the selection of oversized pumps or mixers, increasing operational costs. On the other hand, underestimating viscosity can lead to equipment failure due to insufficient lubrication or suboptimal product quality, such as uneven coatings or unstable emulsions.

Moreover, viscosity is often used to characterize complex fluids and develop predictive correlations for industrial processes. For Newtonian fluids, viscosity remains constant regardless of the applied shear rate, whereas non-Newtonian fluids, such as polymer solutions exhibit varying viscosities depending on the shear rate. These complexities underscore the need for accurate and reliable viscosity data.

Historical Methods for Calculating Viscosity Experimentally

Viscosity, a fundamental fluid property, has historically been measured using various experimental setups. These methods aim to determine the fluid's resistance to flow by quantifying parameters like shear stress and velocity gradient under controlled conditions. Below is a brief overview of some traditional methods

1. Capillary Viscometer

Principle: Measures the time it takes for a liquid to flow through a narrow capillary tube under the influence of gravity or applied pressure. The viscosity is calculated using the Hagen-Poiseuille equation, which relates the flow rate to the viscosity, pressure difference, and dimensions of the capillary.

Applications: Commonly used for low-viscosity Newtonian fluids, such as water and light oils.

Limitations: Accuracy decreases for high-viscosity or non-Newtonian fluids.

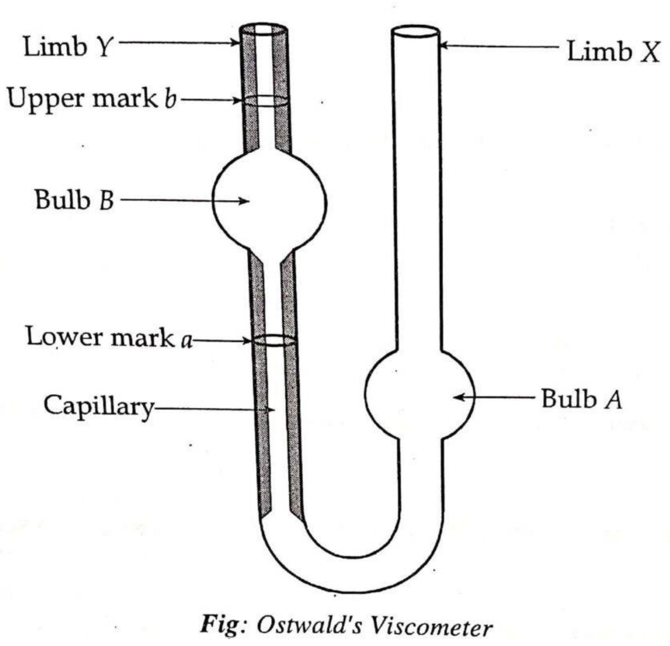


Figure 1. Ostwald's Viscometer

2. Rotational Viscometer

Principle: Measures the torque required to rotate an object (such as a spindle or cylinder) at a constant speed within the fluid. The torque is proportional to the fluid's viscosity.

Applications: Suitable for both Newtonian and non-Newtonian fluids, including paints, coatings, and food products.

Limitations: Requires careful calibration; less effective for very low viscosities.

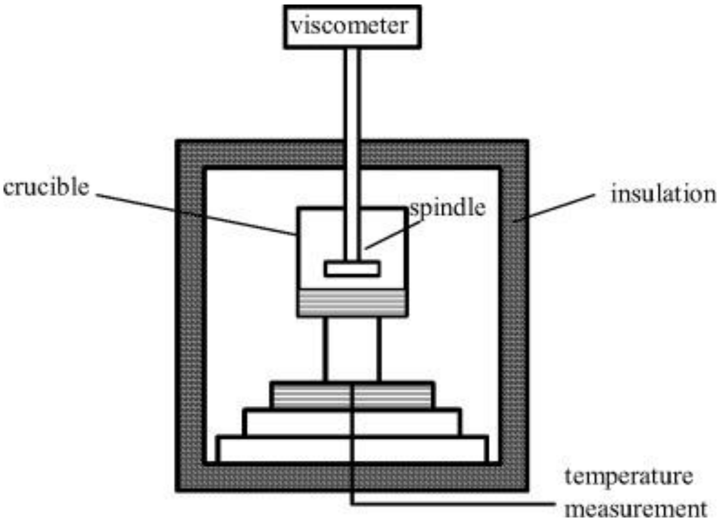


Figure 2. Rotational Viscometer

3. Oscillating Cup Viscometer

Principle: Measures the damping effect on a hollow cylindrical cup oscillating in the fluid. The viscosity is determined based on the energy dissipation caused by fluid resistance.[3]

Applications: Suitable for accurate measurements over a wide temperature range (20–150°C) for low-viscosity liquids.

Advantages: High precision; no calibration required.

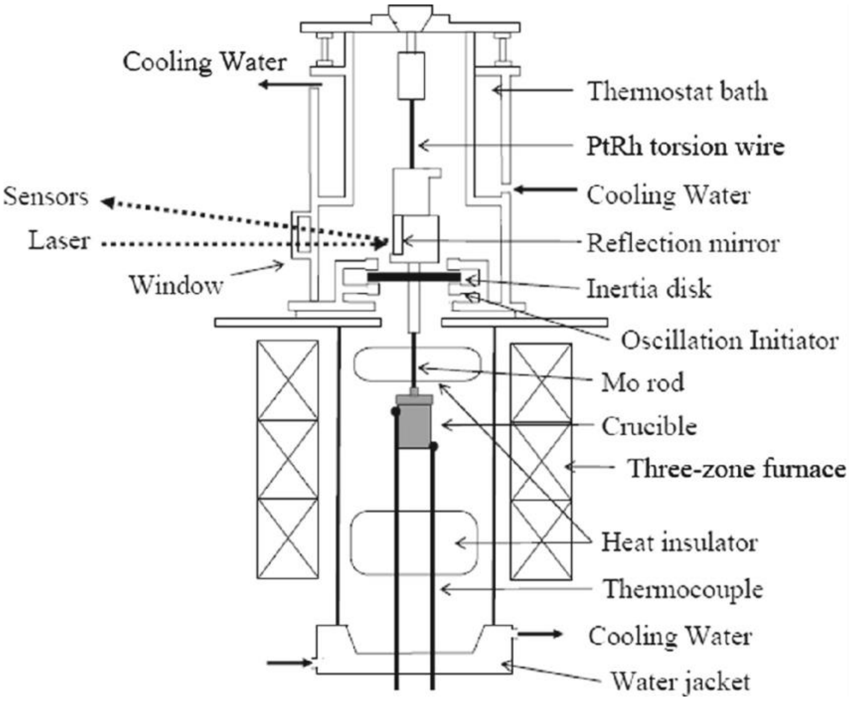


Figure 3. Oscillating Cup Viscometer schematic diagram

Empirical Correlations for Viscosity Prediction: Methods and Limitations

Viscosity correlations are mathematical models developed to estimate the viscosity of liquids based on measurable physical properties such as temperature, pressure, and molecular structure. These correlations simplify the prediction process, especially when experimental data is unavailable, making them invaluable for industrial and engineering applications. While some models are derived from theoretical principles, most rely on empirical data and group contribution methods. Despite their utility, many correlations are limited by their dependency on experimental parameters, reduced accuracy for complex fluids, and inability to handle wide temperature and pressure ranges effectively. Below, we explore three widely used viscosity correlations.

1. Orrick and Erbar (1974) Method

This method uses a group contribution technique to estimate the constants A and B based on the molecular structure of the liquid. It is an adaptation of the Andrade equation, specifically calibrated for organic liquids.

Equation: $ln \frac{\eta_L}{\rho_L M} = A + \frac{B}{T}$

Where, η_L = liquid viscosity, cP,

ρ_L = liquid density at 20 degrees Centigrade, g/cm^3,

M = molecular weight,

T = Temperature, K

Applications:

1. Suitable for a wide range of organic liquids at low temperatures.
2. Effective for non-polar and weakly polar substances.

Limitations:

1. Cannot handle nitrogen- or sulfur-containing compounds.

2. Average error is around 15–16% compared to experimental values.

2. Sastri-Rao Method (1992)

This method calculates the pure liquid viscosity at a given temperature (T) using the relationship between viscosity and vapor pressure and group contributions. Group contributions are determined empirically for different functional groups in the molecule.

Equation: $\eta = \eta_B P_{vp}^{-N}$

Where, η_B is viscosity of that liquid at its boiling point,

P_{vp} is vapour pressure in atmospheres,

N is determined from group contributions.

Applications:

1. Useful for a broad range of temperatures below the boiling point.
2. Particularly effective for hydrocarbons and similar organic liquids.

Limitations:

1. Requires vapor pressure data or accurate prediction models.
2. Errors increase for liquids with complex molecular interactions.

How Machine Learning can improve predictions

Traditional methods for viscosity prediction, such as empirical correlations and group contribution models, rely heavily on experimentally derived parameters, simplifying assumptions, and extensive tabulated data. While these methods have proven effective for many common liquids, they face significant limitations:

Narrow Applicability: Most correlations are tailored for specific classes of fluids (e.g., hydrocarbons or simple polar liquids) and fail for complex systems like mixtures, non-Newtonian fluids, or those with rare functional groups.

Data Dependency: Experimental data for properties such as vapor pressure, critical properties, or molecular structure are often unavailable or challenging to measure.

Reduced Accuracy: At extreme conditions (high/low temperatures or pressures), empirical models often exhibit high deviations from true values.

Inability to Handle Complexity: Correlations struggle with nonlinear interactions between molecular structure, temperature, pressure, and other influencing factors.

Machine Learning (ML) offers a transformative solution to these challenges by leveraging data-driven approaches. Here's how ML models can improve viscosity prediction:

Advantages of ML in Viscosity Prediction

Data-Driven Insights:
ML models learn directly from experimental data, capturing complex and nonlinear relationships that traditional models cannot represent.

Wider Applicability:
ML models can generalize across diverse classes of fluids, including polar and non-polar compounds, mixtures, and even non-Newtonian fluids, given sufficient training data.

Reduction in Experimental Dependency:
By utilizing molecular descriptors, SMILES strings, or readily available physical properties (e.g., molecular weight, temperature), ML models reduce reliance on hard-to-obtain experimental parameters like vapor pressure.

Dynamic Adaptability:

Once trained, ML models can rapidly predict viscosities across a range of conditions (e.g., varying temperatures and pressures) without requiring recalibration.

Improved Accuracy:

By learning from large datasets, ML models can minimize errors in viscosity prediction compared to empirical correlations, especially at extreme conditions or for complex fluids.

Scalability and Automation:

With high computational efficiency, ML models can evaluate large numbers of compounds, enabling rapid screening and optimization in industrial processes.

Objective:

The objective of this project is to develop a Machine Learning (ML)-based model for accurately predicting the viscosities of organic liquids using their thermophysical properties. By leveraging these properties, this project aims to:

1. Create a data-driven model that captures the complex relationships between these properties and liquid viscosity.
2. Minimize reliance on traditional correlations and experimental viscosity measurements, which are often time-consuming and limited in scope.
3. Enable accurate viscosity predictions across diverse classes of organic liquids and varying temperature conditions.
4. Provide a scalable and efficient tool for industrial applications such as process optimization, material development, and product formulation.

This approach emphasizes the integration of thermophysical data with advanced ML techniques to overcome the limitations of existing empirical methods, offering a modern solution to a longstanding challenge in fluid property prediction.

Methodology

In this project, my objective is to develop an accurate and versatile Machine Learning (ML) model for predicting the viscosity of organic liquids based on key thermophysical properties. Recognizing that viscosity is influenced by various molecular and thermodynamic factors, I identified 12 critical properties that serve as the foundation for the model:

1. Normal boiling point and melting point,
2. Surface tension,
3. Refractive index,
4. Critical pressure, temperature, and volume,
5. Dipole moment,
6. Acentric factor,
7. Molecular weight,
8. Heat of vaporization,
9. Density of the liquid.

These properties were chosen because they are either experimentally measurable or widely reported in chemical literature and databases. Together, they capture the intrinsic and extrinsic factors that govern a liquid's viscosity across diverse conditions.

The approach centres on using these 12 features to train ML models capable of identifying complex, nonlinear relationships between the inputs and the target property.

Data Collection

For this project, data collection was a crucial step that involved gathering information from multiple sources to build a comprehensive dataset for viscosity prediction. The key thermophysical properties required for the model, such as critical constants (P_c, T_c, V_c), refractive indices, molecular weight, boiling points, and others, were obtained using the following methods:

Research Papers:

Experimental viscosity data and some related thermophysical properties were collected from published research papers.[1]

These sources provided reliable, experimentally verified data.

Reference Books:

Critical constants like P_c, T_c, V_c were sourced from trusted handbooks such as "Properties of Gases and Liquids"[2]. These resources are widely regarded for their accurate and validated data.

Web Scraping with Selenium:

Properties like refractive indices, molecular weight, and boiling points were extracted from the **PubChem** website using a Python-based web scraping approach.

Web scraping involves automating data collection from websites by parsing their HTML structure.

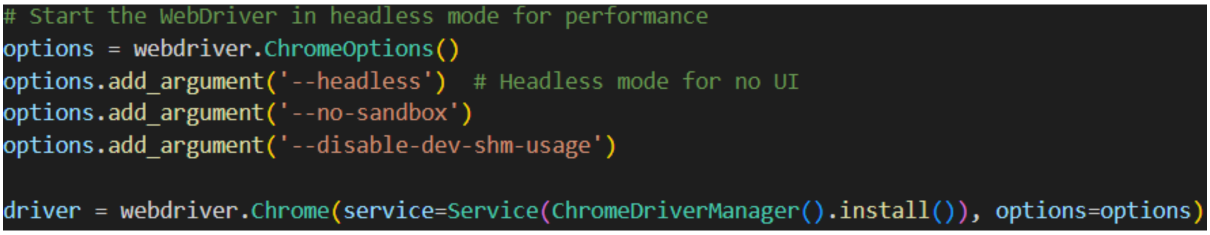
Web Scraping and Selenium

Web Scraping: Web scraping is the process of extracting information from web pages using automation tools. It involves identifying specific data elements within the HTML structure of a webpage (e.g., tags, classes, or IDs) and extracting the desired data programmatically. In this project, molecular weights of compounds, Normal Boiling points, melting points, Density, etc., were fetched from PubChem using a web scraping script.

Selenium: Selenium is an open-source automation framework primarily used for testing web applications. It allows developers to interact with and manipulate web pages programmatically, making it ideal for web scraping dynamic content that requires JavaScript execution.

How Selenium Works:

WebDriver: Selenium uses a WebDriver (e.g., ChromeDriver) to automate browser actions. This WebDriver mimics human interaction with a browser, such as opening pages, clicking elements, or filling forms.

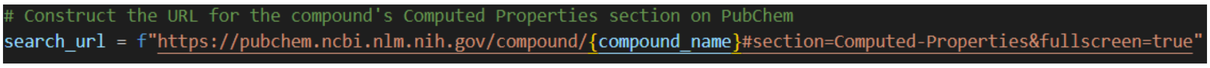


```
# Start the WebDriver in headless mode for performance
options = webdriver.ChromeOptions()
options.add_argument('--headless')  # Headless mode for no UI
options.add_argument('--no-sandbox')
options.add_argument('--disable-dev-shm-usage')

driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)
```

Figure 4. WebDriver Initialization

This piece of code initializes WebDriver in headless mode, which gives better performance and reduced resource usage.

Navigation: Selenium navigates to the desired webpage by constructing URLs based on user input (e.g., compound names for PubChem URLs).

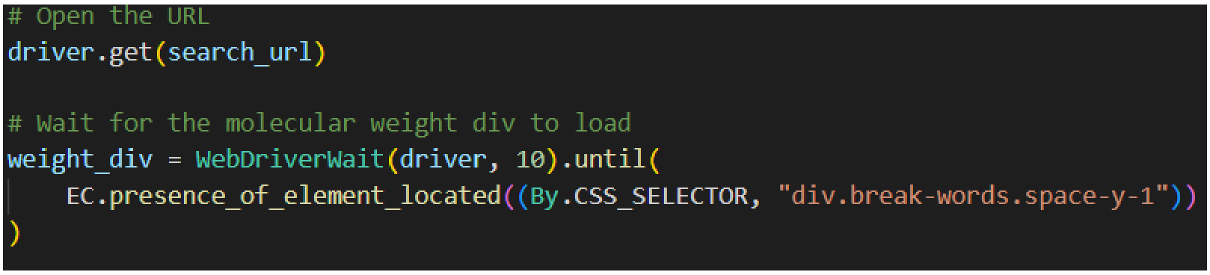


```
# Construct the URL for the compound's Computed Properties section on PubChem
search_url = f"https://pubchem.ncbi.nlm.nih.gov/compound/{compound_name}#section=Computed-Properties&fullscreen=true"
```

Figure 5. Navigation

compound_name is replaced with various compound names to extract the data from this URL.

Element Interaction: It identifies HTML elements using locators like CSS selectors or XPath.

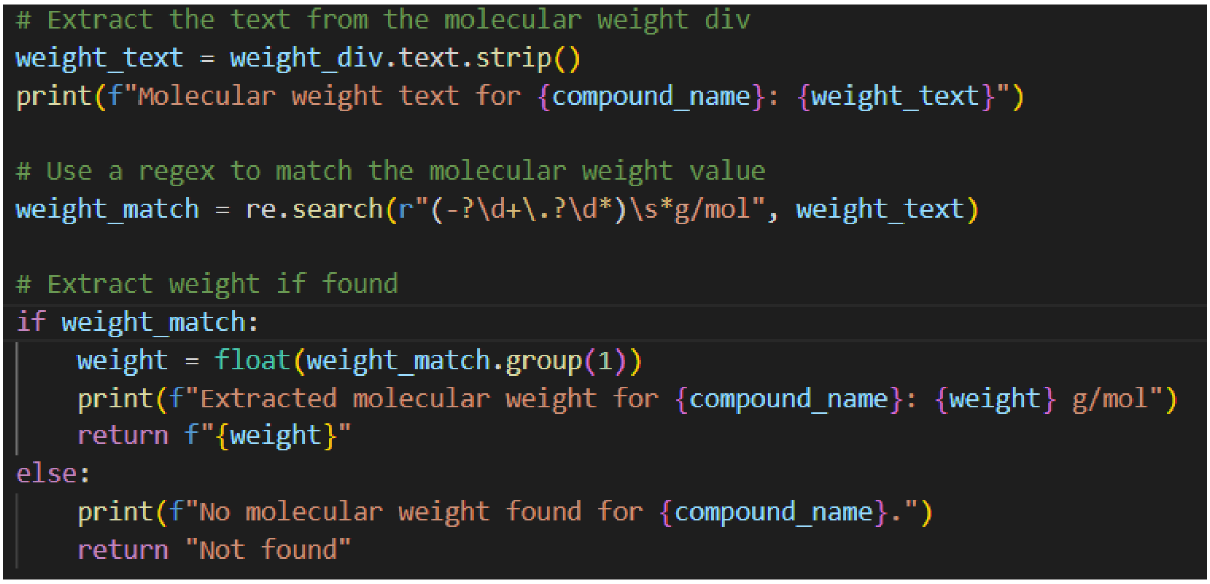


```
# Open the URL
driver.get(search_url)

# Wait for the molecular weight div to load
weight_div = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.CSS_SELECTOR, "div.break-words.space-y-1"))
)
```

Figure 6. Element Interaction by CSS Selector

In this case, the molecular weight section on PubChem was located using CSS selectors, and molecular weight is present in “break-words.space-y-1” div in PubChem page for any compound.

Data Extraction: The text content of the identified element is extracted and processed



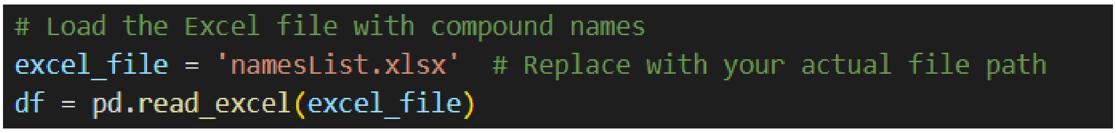
```
# Extract the text from the molecular weight div
weight_text = weight_div.text.strip()
print(f"Molecular weight text for {compound_name}: {weight_text}")

# Use a regex to match the molecular weight value
weight_match = re.search(r"(-?\d+\.?\d*)\s*g/mol", weight_text)

# Extract weight if found
if weight_match:
    weight = float(weight_match.group(1))
    print(f"Extracted molecular weight for {compound_name}: {weight} g/mol")
    return f"{weight}"
else:
    print(f"No molecular weight found for {compound_name}.")
    return "Not found"
```

Figure 7. Data Extraction code snippet

Here, weight_match is true if the weight_text contains a substring with given units of molecular weight (g/mol in this case) and if weight_match is true, the part of string just before the given units is extracted and converted into float values, otherwise “Not Found” is returned. Regex is used for parsing the string. For other properties such as normal boiling point, I have written code for units conversion to keep the data consistent.

For example, if the weight_text is “142.28 g/mol”, the weight_match becomes true and 142.28 is extracted.

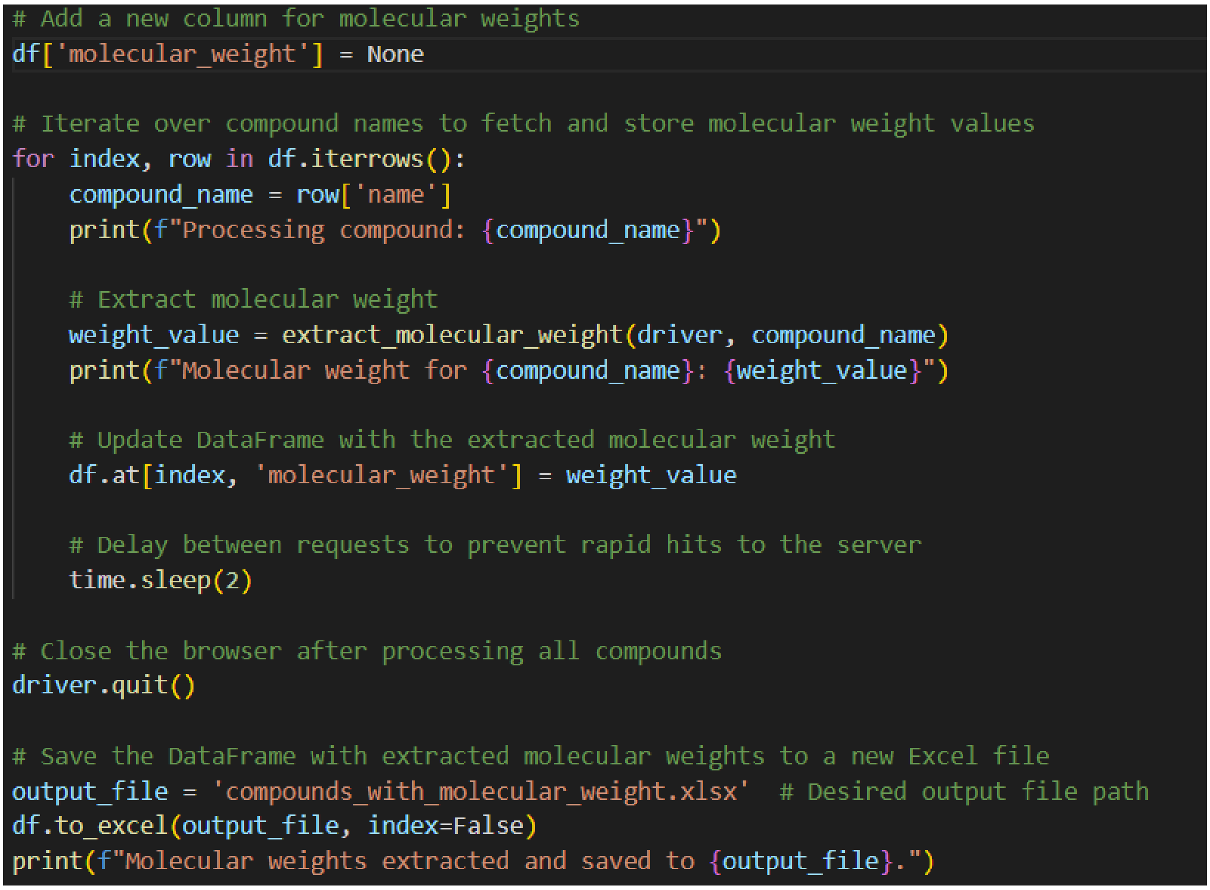
Iteration: Selenium iterates through a list of compounds, performs these steps for each, and appends the results to a dataset.



```
# Load the Excel file with compound names
excel_file = 'namesList.xlsx'  # Replace with your actual file path
df = pd.read_excel(excel_file)
```

Figure 8. Load the input file

namesList.xlsx file contains names of all compounds for which molecular weights are to be extracted.



```
# Add a new column for molecular weights
df['molecular_weight'] = None

# Iterate over compound names to fetch and store molecular weight values
for index, row in df.iterrows():
    compound_name = row['name']
    print(f"Processing compound: {compound_name}")

    # Extract molecular weight
    weight_value = extract_molecular_weight(driver, compound_name)
    print(f"Molecular weight for {compound_name}: {weight_value}")

    # Update DataFrame with the extracted molecular weight
    df.at[index, 'molecular_weight'] = weight_value

    # Delay between requests to prevent rapid hits to the server
    time.sleep(2)

# Close the browser after processing all compounds
driver.quit()

# Save the DataFrame with extracted molecular weights to a new Excel file
output_file = 'compounds_with_molecular_weight.xlsx'  # Desired output file path
df.to_excel(output_file, index=False)
print(f"Molecular weights extracted and saved to {output_file}.")
```

Figure 9. Iteration and data extraction

This code describes the iteration process. For each compound, its name is replaced in the URL, and extract_molecular_weight function extracts molecular weight in the process described above and then the extracted value is stored in another data frame and then converted into excel with the help of pandas library in python. Delay of 2 seconds is used to avoid sending requests rapidly. Exceptions were caught to ensure robust handling of issues like missing data or failed requests. “Not Found” values are manually checked again to correct them. Similar code is used to extract values of other properties like Normal Boiling point, etc.

Benefits of Using Selenium for This Project

Dynamic Content Handling: PubChem pages dynamically load data with JavaScript, which Selenium can execute and interact with, unlike static web scraping tools.

Scalability: Selenium enabled automated processing of multiple compounds, saving significant time compared to manual data collection.

Flexibility: It allowed custom parsing and handling of specific data fields, such as molecular weight in this case.

Data Preprocessing

Data was processed and Exploratory Data Analysis was performed to find out the relationship between input features and target variable viscosity. Brief description of data is as follows

Feature	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
Melting Point (°F)	121	-82.24	84.04	-261.2	-144	-99.6	-9	73
Boiling Point (°F)	121	249.1	104.93	88.7	171.1	241	322	572
Surface Tension (dynes/cm)	121	27.27	6.51	6.58	23.7	26.2	29.87	47.99
Refractive Index	121	1.4295	0.057	1.3292	1.3869	1.4205	1.4601	1.5863
Density (g/cm^3)	121	0.9323	0.2527	0.626	0.7936	0.862	0.995	2.279
Heat of Vaporizat-Ion (kJ/mol)	121	41.59	12.414	20.1	33.06	39.72	45.717	82.145
Molecular weight (g/mol)	121	100.68	35.882	32.04	74.12	98.14	120.15	226.44
Tc (K)	121	583.76	70.44	389	539.79	574.6	632.4	753

Pc (bar)	121	41.2	12.79	14	32	40.6	48.95	80.97
Vc (cm$^{3/}$mol)	121	341.7	156.46	118	237	311	386	1034
Accentric factor	115	0.3853	0.142	0.138	0.281	0.345	0.473	0.795
Dipole Moment (Debye)	113	1.432	1	0	0.4	1.7	1.9	3.8
Viscosity (cP)	121	3.75	13.962	0.24	0.478	0.805	1.804	130.3

Table 1. Description of dataset

Exploratory Data Analysis

Exploratory data analysis was done to explore the nature and intrinsic relationship between the variables. Data was visualized using boxplots, scatter plots, bell curves and some key observations were made as follows

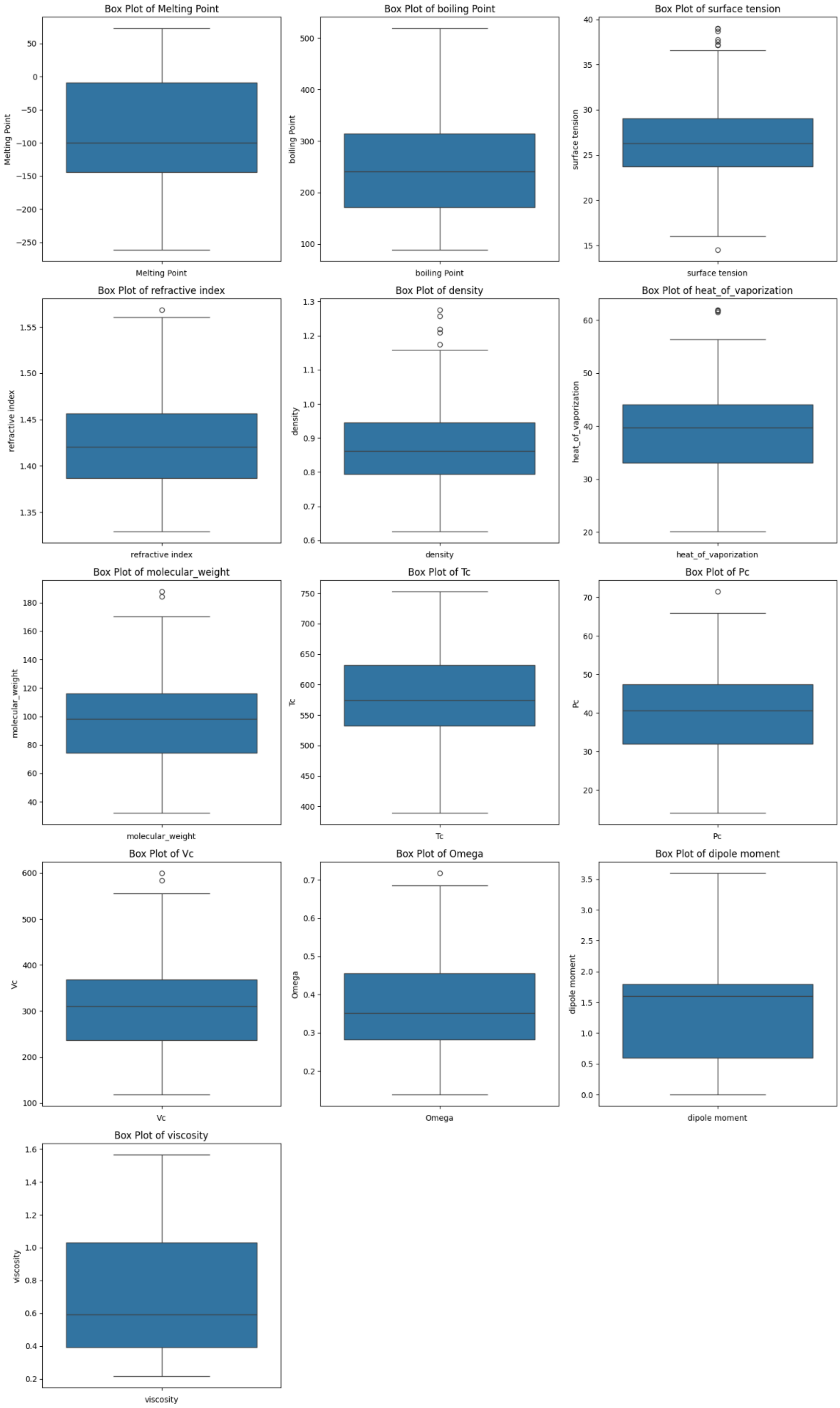


Figure 10. Box plots of all features and target

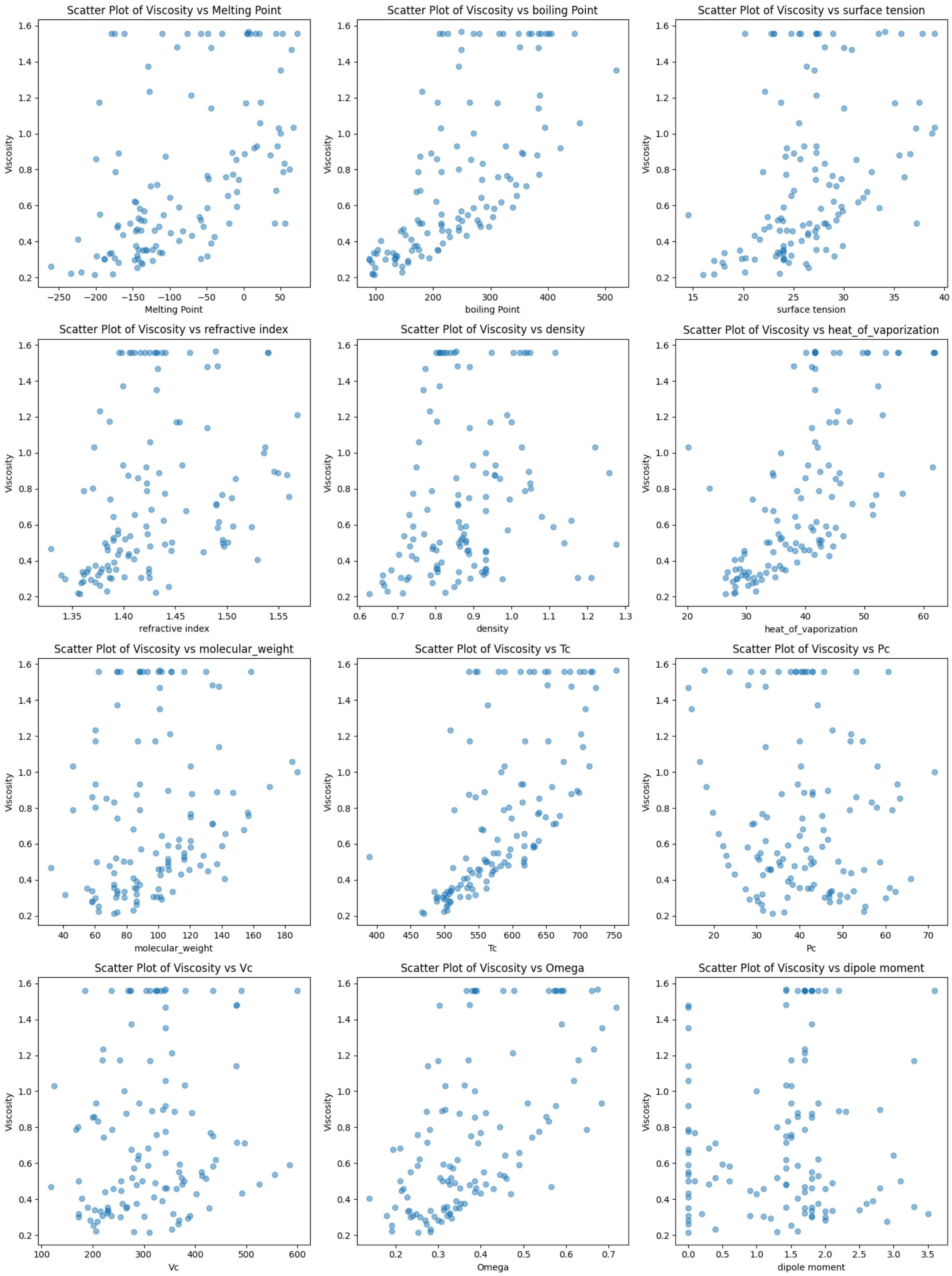


Figure 11. Scattered plot showing relationship between each of the features and target

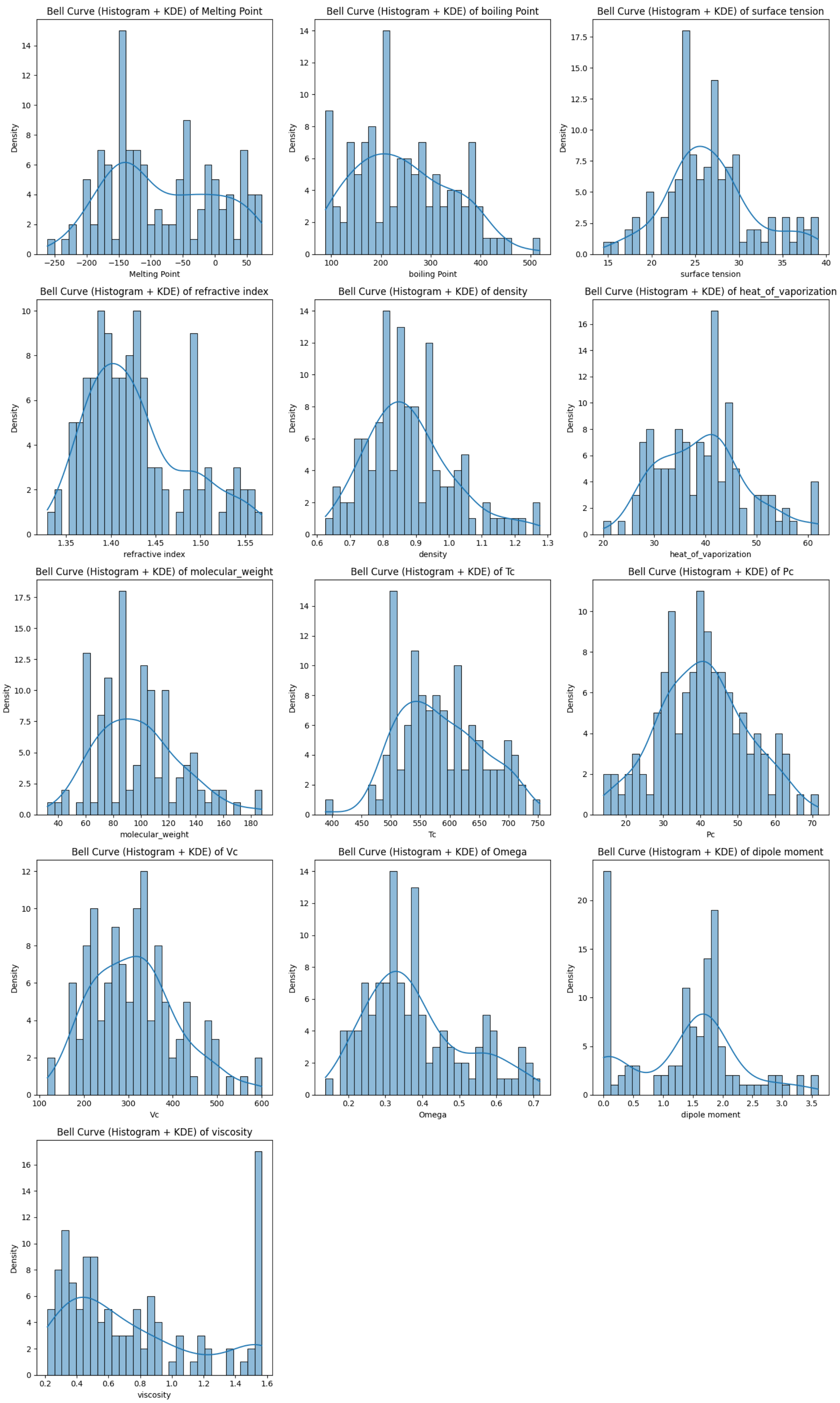


Figure 12. Bell curves for all variables

Key observations made from EDA

The exploratory data analysis revealed several critical insights into the dataset, highlighting areas that require attention during preprocessing and modelling to ensure robust predictions.

From the bell curves (histograms with KDE), it was observed that many features exhibit significant outliers, such as melting point, density, critical volume (V_c), and molecular weight. These outliers, if not handled properly, could skew the model's performance. Additionally, several features, including surface tension, viscosity, and heat of vaporization, displayed moderate to high skewness, suggesting the need for transformations (e.g., log or Box-Cox transformations) to normalize their distributions. While some features like refractive index and critical temperature (T_c) had symmetric distributions, others such as omega (Accentric factor) and dipole moment showed irregular or multi-modal patterns, possibly reflecting diverse chemical families within the dataset.

The scatter plots (features vs. viscosity) highlighted the absence of clear linear relationships between most features and the target variable, viscosity. For instance, while critical temperature (T_c) and omega showed some discernible patterns, most relationships were non-linear and noisy, necessitating the use of advanced ML algorithms (e.g., neural networks or tree-based methods) to capture these complex interactions. Additionally, clustering behaviour was observed in features like surface tension and boiling point, indicating potential subgroups such as polar and non-polar liquids. Moreover, heteroscedasticity was evident in some scatter plots (e.g., density and dipole moment), with variability in viscosity predictions increasing for certain ranges of the features.

The box plots further confirmed the presence of outliers across nearly all features, especially in molecular weight, surface tension, V_c, and boiling point. These outliers could represent unique compounds or measurement errors, requiring domain knowledge for proper handling. The range of feature values varied significantly, with some features like boiling point and critical temperature having wide distributions (88–753), while others like omega and density were relatively constrained. This disparity highlights the importance of feature scaling to ensure all features contribute equally to model training.

As a result of the exploratory data analysis, it was observed that the viscosity feature exhibited significant skewness (greater than 0.5). To address this, a log transformation was applied to normalize its distribution. Additionally, feature scaling was implemented to ensure that all variables contributed equally to the model. Outliers were handled using the IQR method, which effectively replaced extreme values with more representative ones, while missing values in features like dipole moment and omega were replaced with mean values. These preprocessing steps ensured a clean and well-prepared dataset for building a robust machine learning model.

Model Selection

The dataset's inherent non-linear relationships necessitated the use of advanced machine learning models capable of capturing intricate patterns and complex feature interactions. To achieve accurate viscosity predictions, three distinct models were selected: XGBoost, a stacked ensemble combining XGBoost and LightGBM, and a neural network using a Multi-Layer Perceptron (MLP). Each model was carefully chosen based on its unique capabilities, trained rigorously, and optimized to enhance prediction accuracy. Below is a detailed explanation of each model, their underlying logic, and the steps taken to optimize their performance.

1. XGBoost (Extreme Gradient Boosting)

Overview:

XGBoost, an implementation of gradient boosting, builds decision trees iteratively, where each tree focuses on correcting the residual errors made by previous ones. The model minimizes a loss function, such as mean squared error, by leveraging gradient descent to adjust predictions. Unlike traditional gradient boosting, XGBoost introduces regularization techniques (L1 and L2 penalties) to prevent overfitting, making it robust and highly efficient for real-world datasets. Its ability to handle missing values, feature importance ranking, and parallel processing further makes it a versatile choice for non-linear and complex datasets like this one.

How It Works:

The algorithm begins with an initial prediction (e.g., the mean value of the target). At each iteration, a new decision tree is trained to minimize the residual errors from the previous predictions. Predictions from all trees are combined using a weighted sum to make the final prediction. Regularization terms in the objective function ensure that the model does not overfit by penalizing overly complex trees.

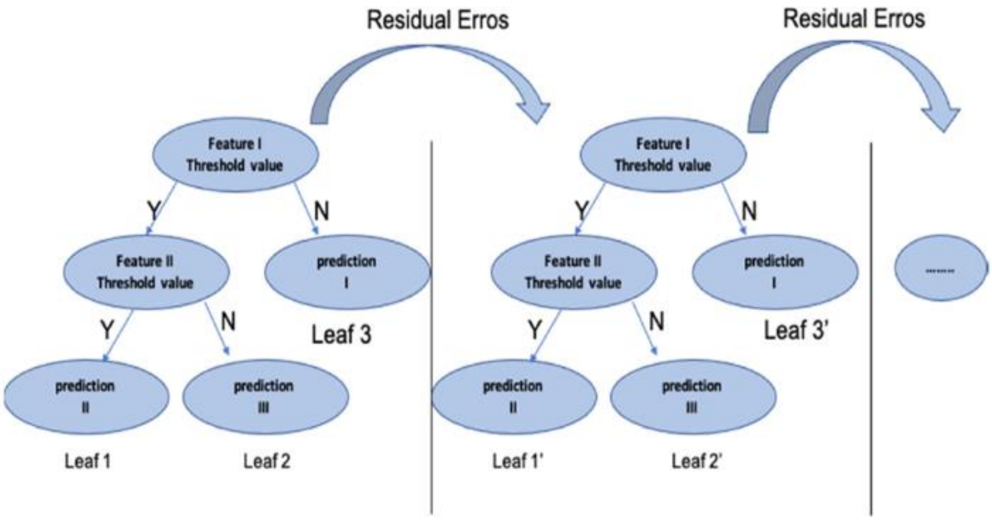


Figure 13. Figure illustrating XGBoost

Training and Optimization:

To extract the best performance from XGBoost, hyperparameters were optimized using Optuna, an automated hyperparameter tuning framework. Critical parameters tuned included:

1. **n_estimators**: Number of trees in the model, controlling model complexity.
2. **max_depth**: Maximum depth of each tree, regulating feature interactions.
3. **learning_rate**: Step size for weight updates, balancing convergence speed and accuracy.
4. **subsample and colsample_bytree**: Fractions of samples and features used to train each tree, reducing overfitting.
5. **reg_alpha and reg_lambda**: L1 and L2 regularization parameters for further control of overfitting.

The model was trained using cross-validation to ensure consistent performance across different splits of the data. XGBoost provided a solid baseline with strong predictive capabilities and a well-balanced bias-variance trade-off.

2. Stacked Ensemble Model (XGBoost + LightGBM with Ridge Regressor)

Overview:

Stacked ensemble models improve accuracy by combining the strengths of multiple base learners. In this project, XGBoost and LightGBM were chosen as the base models due to their complementary strengths. XGBoost excels in modeling complex feature interactions and handling outliers. LightGBM (Light Gradient Boosting Machine) is faster and more efficient, particularly when dealing with large datasets, thanks to its histogram-based approach for feature selection and split finding.

The Ridge regression model served as the meta-learner, combining predictions from the base models into a final, more accurate prediction. Ridge regression adds an L2 penalty to the cost function, reducing the impact of multicollinearity and ensuring smooth generalization.

How It Works:

The two base learners, XGBoost and LightGBM, were trained independently on the same dataset. Each model made predictions on the training and test sets. These predictions were used as new features for training the Ridge regression meta-learner. Ridge regression learned to optimally combine the predictions from XGBoost and LightGBM, minimizing overall error and improving robustness.

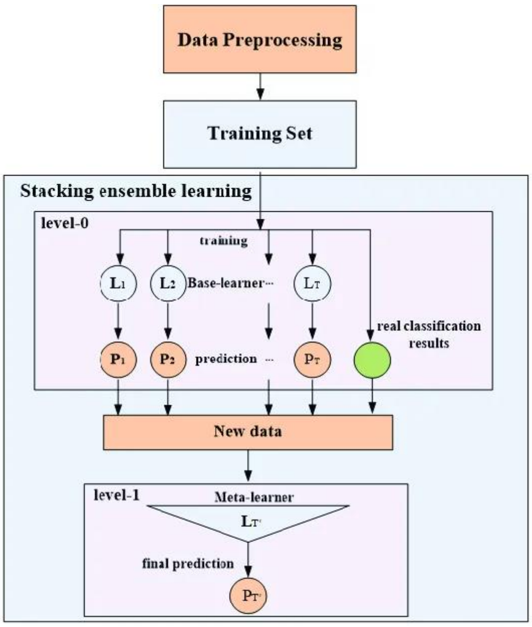


Figure 14. Figure Illustrating general stacked ensemble model

Above image clearly illustrates the concept of a stacking ensemble learning model. For this case, Base-learner is XGBoost and LightGBM, meta-learner is Ridge Regressor.

Training and Optimization:

Both XGBoost and LightGBM were optimized separately. XGBoost's hyperparameters were fine-tuned using Optuna, as described earlier. LightGBM's key parameters, such as num_leaves (maximum leaf nodes per tree), max_depth (tree depth), and learning_rate, were manually adjusted for better performance. The stacked ensemble was evaluated using five-fold cross-validation to ensure that combining the base models consistently improved accuracy. By leveraging the complementary strengths of XGBoost and LightGBM, the ensemble delivered better predictions than any individual model, achieving lower error rates and higher R^2 scores.

3. Neural Networks (Multi-Layer Perceptron)

Overview and Logic:

Neural networks are powerful algorithms designed to approximate complex functions by learning non-linear patterns directly from data. A Multi-Layer Perceptron (MLP), a type of feedforward neural network, consists of an input layer (features), hidden layers (neurons), and an output layer (target prediction). Each neuron applies a weighted sum of its inputs followed by a non-linear activation function, such as ReLU, allowing the network to model intricate relationships. MLPs are particularly effective for datasets where the relationships between features and the target are highly non-linear and multi-dimensional.

How It Works:

Data flows through the network layer by layer, with each layer learning a progressively higher level of abstraction. The network's weights are adjusted iteratively using backpropagation, which computes gradients of the loss function (e.g., mean squared error) with respect to the weights. These gradients guide weight updates through gradient descent, minimizing the loss and improving predictions over time.

Training and Optimization:

To maximize the neural network's performance, hyperparameters were optimized using Optuna. Key parameters included:

Hidden layer sizes: Number of neurons in each layer, controlling the network’s capacity.

Number of layers: Total depth of the network, affecting its ability to model complex patterns.

Alpha: Regularization parameter to prevent overfitting by penalizing large weights.

Learning rate: Step size for updating weights during training, balancing convergence speed and accuracy.

Max iterations: Number of iterations for training, ensuring the model converges.

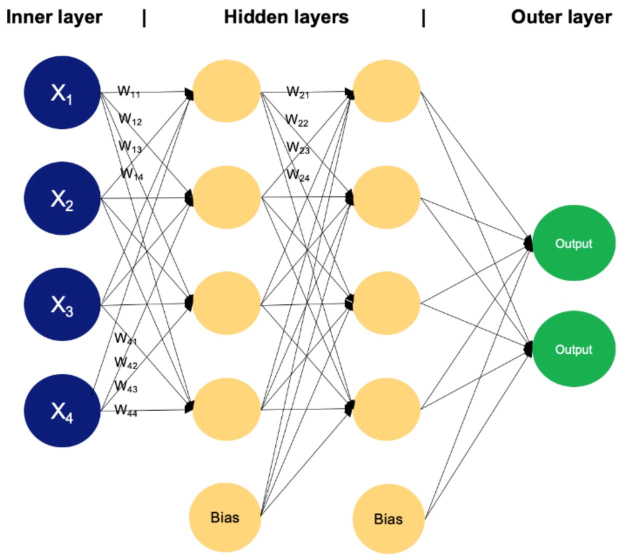


Figure 15. Example of MLP with 2 layers

The model was trained using five-fold cross-validation to evaluate its performance across different subsets of the data. Regularization and early stopping techniques were employed to prevent overfitting, ensuring the network generalizes well to unseen data. By combining flexibility with careful tuning, the neural network achieved competitive performance, especially for highly non-linear relationships.

Results

The results obtained by these 3 models are tabulated in the table below.

Model	MAPE	R2	RMSE	MAE
XGBoost	0.1414	0.655	0.67	0.255
Stacked Ensemble	0.1651	0.8677	0.4296	0.2303
MLP	0.1068	0.9446	0.278	0.1485

Table 2. Performance of various models

Scattered plots of actual vs predicted values for these 3 models are shown below.

1. XGBoost

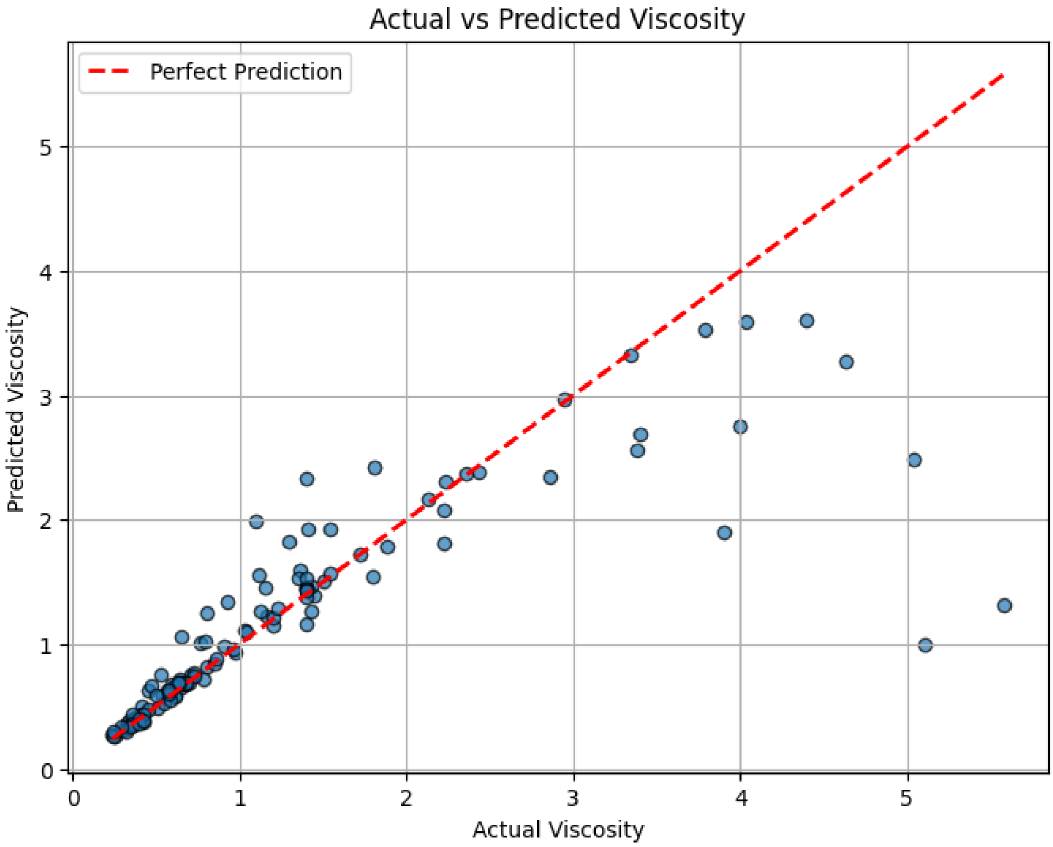


Figure 16. Predicted vs Actual values for XGBoost model.

2. Stacked Ensemble Model

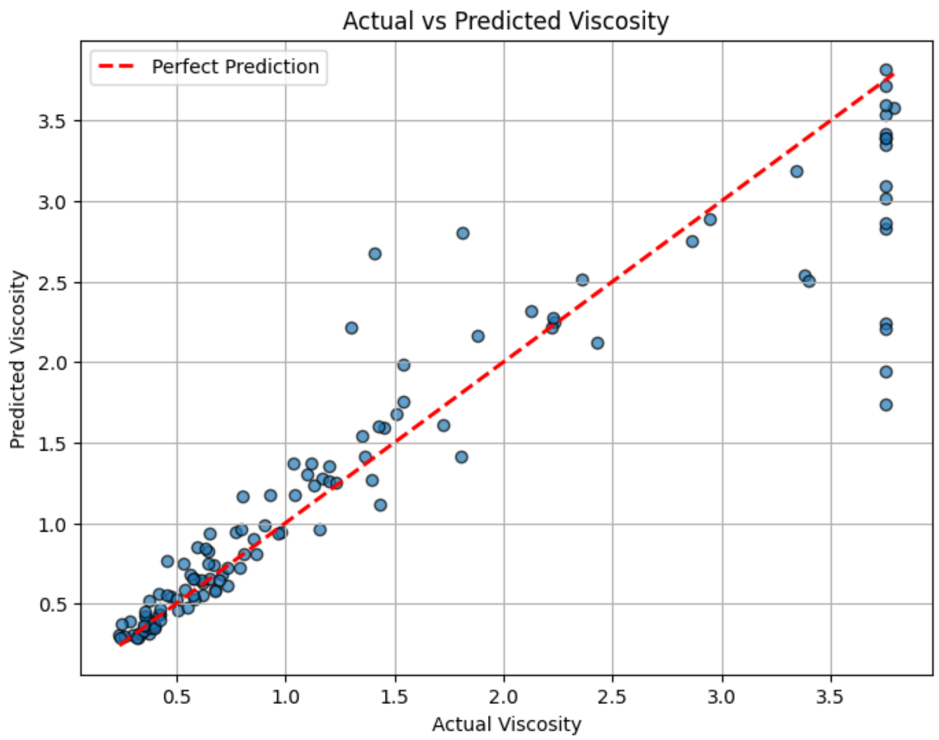


Figure 27. Predicted vs Actual values for Stacked Ensemble Model.

3. MLP

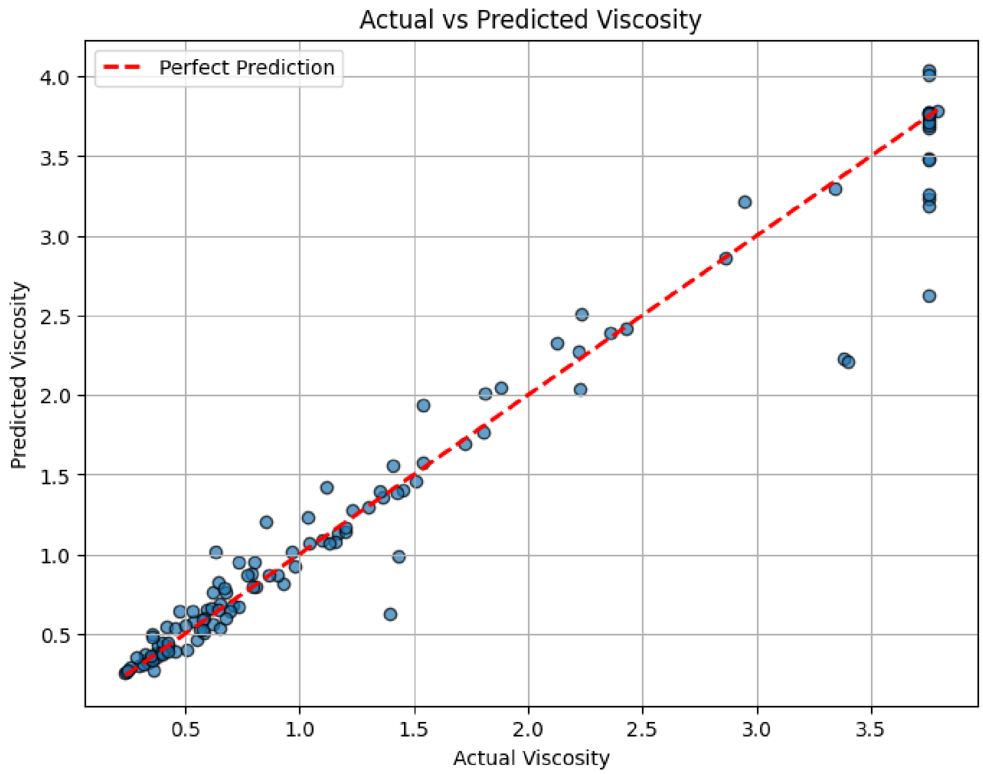


Figure 18. Predicted vs Actual values for MLP Model.

Following table shows MAPE in prediction of a small sample of data,

Compound Name	Orrick and Erbar	Sastri and Rao	Przezdziecki and Sridhar	XGB	Stacked Ensemble	MLP
Acetone	-9.4	1.6	-1.2	-6.48	-9.75	-2.4
Benzene	-35	-6.6	7.3	63.14	43.63	5.26
Chloroform	34	5.7	-8.1	8.27	-4.98	-9.14
Carbon Tetrachloride	22	-2	-15	-0.26	-3.21	4.36
Cyclohexane	-51	29.7	-38	9.53	-10.66	-2.01
O-Xylene	5	-4.5	-4.8	1.89	-0.32	-1.06
Heptane	-3.3	-1	-17	20.36	34.4	-5.14
MAPE	22.8	7.3	13.05	15.7	15.27	4.195

Table 3. Comparision of Existing correlations with Models performance

Observations

The results obtained through the three models XGBoost, Stacked Ensemble, and Multi-Layer Perceptron (MLP) show significant differences in their predictive performance when evaluated using metrics such as Mean Absolute Percentage Error (MAPE), R^2 (coefficient of determination), RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error). Each model exhibited distinct strengths and weaknesses, with clear trends emerging from the analysis.

XGBoost provided moderate predictive performance, achieving an R^2 of 0.655, which indicates that 65.5% of the variance in viscosity was explained by the model. With a MAPE of 0.1414 and an RMSE of 0.67, XGBoost demonstrated reasonable accuracy but struggled with certain data points. The scatter plot for XGBoost highlights its limitations, showing noticeable deviations from the ideal prediction line for some compounds. This suggests that while XGBoost captures some patterns in the data, it may have difficulty handling complex relationships inherent in viscosity predictions.

The Stacked Ensemble model showed substantial improvement over XGBoost by combining predictions from XGBoost and LightGBM through a Ridge meta-regressor. This approach yielded an R^2 of 0.8677, explaining 86.77% of the variance in viscosity values. The RMSE of 0.4296 reflects improved precision, and the scatter plot shows tighter clustering around the perfect prediction line. However, the MAPE of 0.1651 indicates that the model still struggles with variability in predictions for certain compounds. This result underscores the strength of ensemble learning in leveraging multiple base models to improve accuracy and generalization.

The Multi-Layer Perceptron (MLP) neural network emerged as the best-performing model, achieving an R^2 of 0.9446, which indicates that it explained 94.46% of the variance in viscosity. Its RMSE of 0.278 and MAE of 0.1485 were significantly lower than those of the other models, demonstrating the MLP's ability to capture complex non-linear relationships. The scatter plot for MLP shows near-perfect alignment with the ideal prediction line, highlighting the precision and reliability of the model. The MAPE of 0.1068 further establishes MLP as a superior choice for viscosity prediction, with significantly lower errors than both XGBoost and the Stacked Ensemble.

The scatter plots further illustrate the models' performance. XGBoost showed greater deviations from the ideal prediction line, particularly for outliers, while the Stacked Ensemble reduced these deviations by leveraging multiple models. MLP, on the other hand, showed the strongest alignment with the ideal line, demonstrating its ability to generalize well across the dataset.

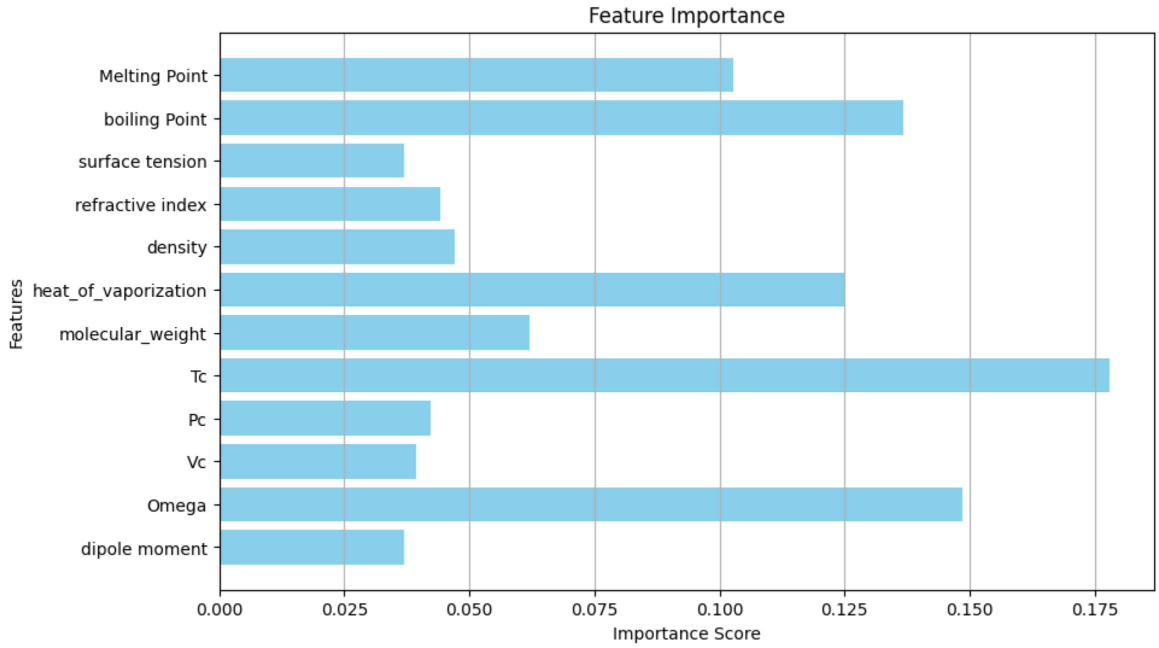


Figure 19. Plot showing feature importance for predicting viscosity

The feature importance plot highlights the relative influence of various thermophysical properties on the prediction of viscosity. Among the features, critical temperature (Tc), boiling point, and omega (acentric factor) exhibit the highest importance scores, signifying their critical role in determining viscosity. These findings are consistent with the theoretical understanding of viscosity, where parameters linked to molecular interactions and phase behavior (like Tc and boiling point) are highly influential. Other features, such as heat of vaporization and molecular weight, also have notable contributions, reflecting their relevance to the intermolecular forces and the overall molecular size.

Features such as surface tension, refractive index, and dipole moment show comparatively lower importance, which might indicate that their relationship with viscosity is weaker or that their effect is already captured indirectly through the more dominant features. Interestingly, density has a moderate influence, aligning with its role in characterizing the physical state of liquids.

Despite these insights, some potentially significant features might have been overlooked, which could capture the molecular interactions and structural complexities more effectively:

Molecular Geometry or Shape Descriptors: Parameters like sphericity or aspect ratio could help better account for the effects of molecular shape on viscosity.

Hydrogen Bonding Contribution: Including quantitative metrics of hydrogen bonding could significantly improve the model for liquids where such interactions dominate.

Electrostatic Properties: Features like polarizability or charge distribution might capture additional nuances in intermolecular forces.

Functional Group Contributions: Explicit representation of functional groups (e.g., alcohols, ethers) might help in modeling more specific molecular behaviors.

The industrial implications of these findings are substantial. Accurate viscosity prediction is crucial for optimizing processes in industries such as petrochemicals, pharmaceuticals, and polymers. The analysis demonstrates that ML-based approaches, particularly MLP, offer significant improvements over traditional correlations. Traditional methods like those by Orrick and Erbar, Sastri and Rao, and Przezdziecki and Sridhar showed varying accuracies for the same compounds, with MAPE ranging from 7.3% to 22.8%. In contrast, the MLP model achieved a MAPE of just 4.195% on a sample dataset, showcasing its reliability and robustness. This difference is particularly evident in challenging compounds like benzene and heptane, where traditional methods struggled to provide accurate predictions.

In summary, while the current feature set provides a robust basis for viscosity prediction, incorporating additional descriptors that account for molecular size, shape, and specific interactions could further enhance the model's predictive performance.

Conclusion

The study undertaken focuses on the prediction of viscosity, a critical property of organic liquids, using advanced machine learning techniques. Viscosity plays an essential role in industries such as chemical manufacturing, pharmaceuticals, petrochemicals, and material design, directly affecting processes like mixing, transport, and heat transfer. Traditional experimental methods for measuring viscosity, such as capillary viscometers and rotational rheometers, though accurate, are time-consuming, expensive, and limited to specific operational conditions. Empirical correlations like those of Orrick and Erbar, Sastri and Rao, and Przezdziecki and Sridhar, although widely used, often fail to capture the complex relationships inherent in diverse organic liquids. They are also limited in accuracy when applied to polar and non-polar liquids, mixtures, or compounds with intricate molecular structures. In this study, we attempted to address these challenges by leveraging machine learning approaches.

A comprehensive dataset was developed by collecting data from various sources, including research publications, technical handbooks, and web scraping techniques. The dataset included 12 critical thermophysical properties such as boiling point, melting point, surface tension, molecular weight, refractive index, critical constants, acentric factor, dipole moment, heat of vaporization, and density, which were selected for their direct or indirect influence on viscosity. To handle missing data and outliers, iterative imputation and the replacement of outliers with mean values were employed. The viscosity data, found to exhibit skewness, was log-transformed to normalize its distribution. Exploratory data analysis revealed the non-linear and complex relationships between the input features and viscosity, reinforcing the need for advanced machine learning models to capture these intricate patterns.

Three machine learning models were implemented: XGBoost, a Stacked Ensemble model combining XGBoost and LightGBM, and a Multi-Layer Perceptron (MLP). XGBoost demonstrated reasonable predictive accuracy, achieving an R^2 of 0.655 and a MAPE of 0.1414. While XGBoost effectively captured some patterns, its limitations became evident in the scatter plots, where deviations from the ideal prediction line highlighted its struggle with complex relationships in the data. The Stacked Ensemble model, combining XGBoost and LightGBM with a Ridge meta-regressor, improved upon XGBoost, achieving an R^2 of 0.8677 and a MAPE of 0.1651. This ensemble approach effectively leveraged the strengths of both base learners, reducing error and enhancing generalization, as observed in the tighter clustering of predictions around the perfect prediction line.

The MLP neural network emerged as the most effective model, with an R^2 of 0.9446, RMSE of 0.278, and MAPE of 0.1068. Its ability to capture complex, non-linear relationships was evident in its superior alignment with the ideal prediction line in scatter plots. The MLP's superior performance can be attributed to its ability to learn intricate dependencies in the data through multiple layers and neurons. The feature importance analysis revealed that properties like critical temperature, boiling point, and molecular weight played a significant role in predicting viscosity, aligning with theoretical and experimental understandings of molecular interactions.

However, it is important to recognize potential sources of error in this study. The features used, while comprehensive, may not fully account for all molecular and thermodynamic properties influencing viscosity. For example, descriptors capturing molecular shape, size distribution, or intermolecular forces could provide additional predictive power. Additionally, the dataset's size and diversity might limit the models' generalizability to novel compounds or extreme conditions. Future work could focus on incorporating more sophisticated molecular descriptors, expanding the dataset, and exploring hybrid models that integrate domain knowledge with data-driven approaches.

The comparison with traditional correlations highlighted the substantial advantages of ML models. While empirical methods achieved MAPE values ranging from 7.3% to 22.8% for various compounds, the ML models, particularly MLP, significantly outperformed them with a MAPE of just 4.195% for a sample dataset. This improvement was especially pronounced for challenging compounds like benzene and heptane, where traditional methods often yielded large errors. The industrial implications of these findings are profound. Accurate viscosity prediction through ML can optimize processes, reduce reliance on costly experimental methods, and enhance efficiency in industries ranging from petrochemicals to pharmaceuticals. The feature importance analysis further aids in understanding the relative influence of different properties, guiding experimentalists and engineers in prioritizing measurements or estimations of critical features.

In conclusion, this study demonstrates the transformative potential of machine learning in addressing the longstanding challenge of viscosity prediction. The MLP model, in particular, offers a robust and scalable solution, outperforming both traditional correlations and other ML approaches. By integrating advanced machine learning techniques with thermophysical property data, this work provides a foundation for more efficient and accurate viscosity prediction, paving the way for innovations in industrial applications and process optimization.

References

[1] *Estimation of the Liquid Viscosity of Organic Compounds with a Quantitative Structure-Property Model*, by Ovidiu Ivanciuc,, Teodora Ivanciuc, Petru A. Filip, and Daniel Cabrol-Bass,

[2] *Properties of Gases and Liquids*, fifth edition, by Bruce E. Poling, John M. Prausnitz, John P. O'Connell

[3] *Viscosity of Pure Hydrocarbons*, by Boerge Knapstad, Per A. Skjoelsvik and Harald A. Oeye

[4] National Library of Medicine*, National Centre for Biotechnology Information, PubChem,* (https://pubchem.ncbi.nlm.nih.gov/)

[5] *A scalable and integrated machine learning framework for molecular properties prediction,* by Guzhong Chen, Zhen Song, Zhiwen Qi, Kai Sundmacher

[6] *Machine learning for predicting thermodynamic properties of pure fluids and their mixtures,* by Yuanbin Liu, Weixiang Hong, Bingyang Cao.