UG PROJECT REPORT

TITLE: POLYMER RHEOLOGY

New Polymer Rheology Models Based On Machine Learning

Mentored By: Prof. Himanshu Sharma

Submitted By:
Achyut Verma (170041)
Anshika Verma (170131)

Project Goals:

- To explore the insights on the *Application of AI (Artificial Intelligence) in Chemical Engineering*.
- To understand and apply the existing Machine Learning based models on predicting polymer rheology.
- To develop new models to predict the viscosity of HPAM Polymer used in Enhanced Oil Recovery (EOR).

Abstract

Polymer flooding is now considered a technically- and commercially-proven method for enhanced oil recovery (EOR). The viscosity of the injected polymer solution is the key property for successful polymer flooding. Given that the viscosity of a polymer solution has a non-linear relationship with various influential parameters (molecular weight, degree of hydrolysis, polymer concentration, cation concentration of polymer solution, shear rate, temperature) and that measurement of viscosity based on these parameters is a time-consuming process, the range of solution samples and the measurement conditions need to be limited and precise. Viscosity estimation of the polymer solution is effective for these purposes. In this communication, various machine learning algorithms such as Artificial Neural Network (ANN), Random Forest (RF) and Support Vector Machine (SVM) were employed to model the viscosity of HPAM solutions. Then, the accuracy and reliability of the developed models in this study were investigated through graphical and statistical analyses, trend prediction capability, outlier detection, and sensitivity analysis. As a result, it has been found that the ANN model gives the most reliable results with determination coefficients (R^2) more than 0.98 and Root Mean Square Error (RMSE) less than 50. Finally, the suggested models in this study can be applied for efficient estimation of aqueous solutions of HPAM polymer in simulation of polymer flooding into oil reservoirs.

Content

Introduction

Application of AI in Chemical Engineering

A major shortcoming of traditional strategies is the fact that solving chemical engineering problems due to the *highly nonlinear behavior* of chemical processes is often impossible or very difficult. Today, artificial intelligence (AI) techniques are becoming useful due to simple implementation, easy designing, generality, robustness and flexibility. The AI includes various branches, namely, artificial neural networks, fuzzy logic, genetic algorithms, expert systems and hybrid systems. They have been widely used in various applications of the chemical engineering field including modeling, process control, classification, fault detection and diagnosis. AI, particularly ML, plays an important role in generation, use, and management of massive amounts of diverse data, information, and knowledge.

Polymer Rheology

In the primary recovery stage, oil can be mainly produced using natural energy sources, such as reservoir pressure. As oil production progresses, reservoir pressure decreases, and economical oil production is no longer feasible. To reduce the decline of reservoir pressure and push the oil in the reservoir to the production well, water flooding (secondary recovery), which is the process of injecting water into the reservoir, has been commonly applied to most oil fields. The primary and water flooding methods typically extract no more than 10%–40% of the original-oil-in-place [1]. EOR has been proven successful in producing much of the remaining oil and improving the ultimate recovery factor.

Polymer flooding, which is a technique for enhanced oil recovery (EOR), is the process of injecting a viscous polymer solution into the reservoir. Polymer flooding is now considered to be a technically- and commercially-proven EOR method, especially since its large-scale application at Daqing oil field in northern China, which produced about 300,000 barrels of incremental oil per day [2]. Therefore, polymer flooding applications have seen a progressive surge in science and engineering in recent years. A polyacrylamide-based polymer (e.g. HPAM) is one of the most extensively used polymers in chemical EOR processes due to their good solubility in water and cost-effective affordability. As a definition, HPAMs are synthesized from acrylamide monomers; thus, they will have negatively charged linear chain macromolecules, in which some of their monomers are hydrolyzed by an electrolyte [3]. The large viscosities of HPAM solutions are caused by repulsion forces amongst the negatively charged polymer chains, which are dependent on chain extension and degree of acrylamide hydrolysis [4].

The viscosity of the polymer solution is the key property for the successful application of polymer flooding. Polymers are generally non-Newtonian and the **viscosity** is a function of *shear rate*, *polymer* *concentration*, *salinity*, *hardness*, *temperature*, and *molecular weight*. In this study, we will employ various machine learning techniques namely, Artificial Neural Network, Random Forest, Support Vector Machine (SVM) to model the viscosity of a HPAM Polymer (Floppam 3330S).

Project Motivation

What

The *objective* of this study is *to develop new predictive models to predict the viscosity of selective polymers used extensively in enhanced oil recovery for eventual implementation into a chemical flooding numerical reservoir simulator (e.g. UTCHEM)*

Why

A priori estimation of polymer rheology using robust modeling techniques is important for design of polymer floods and prediction using numerical reservoir simulators.

How

The study uses a database of existing experimental data for the rheology of polymers and uses a combination of fundamental, physical models and machine learning methods to develop new predictive models.

Dataset Description

The data includes the measured polymer viscosity at various shear rates, polymer concentrations, brine salinity (NaCl concentration), hardness (Ca2+ Concentration) and temperature.

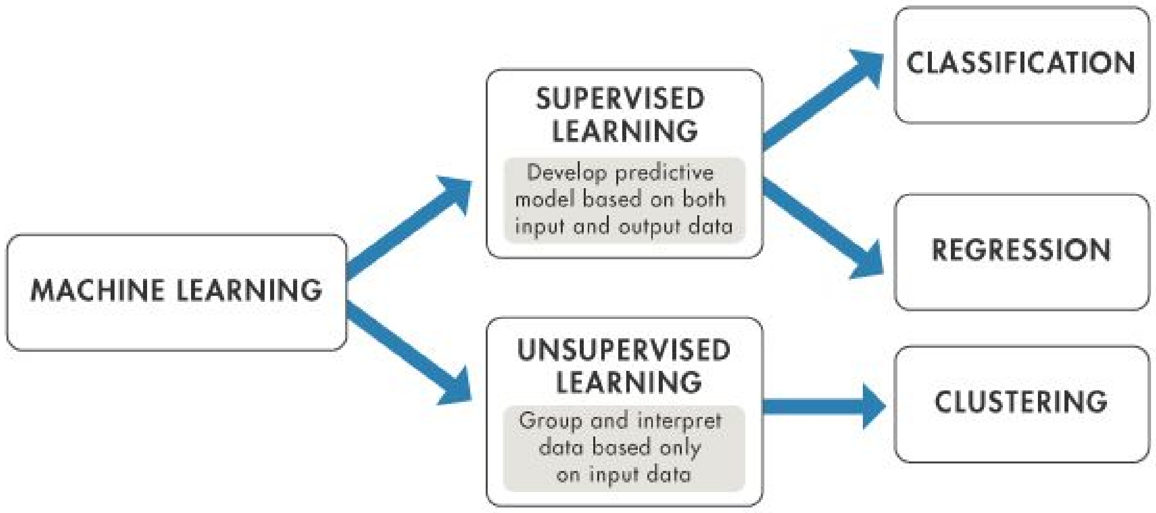
Modeling

The study explores advanced machine learning algorithms, such as Random Forests, and Artificial Neural Networks and Support Vector Machines to develop more robust and reliable models to predict viscosity at a given set of variables (concentration, salinity, hardness, etc).

Machine Learning

Machine learning, sometimes referred to as statistical learning, is an automated data analysis process that attempts to understand the data and identify possible connections between the data features. Statistical learning can be interpreted as a set of tools used to understand the data. In addition, the automation part of the machine learning's algorithm, which includes learning from data, building and updating the model and establishing prediction, is a satisfactory feature of a machine learning model.

Machine learning problems fall in one of two types of learning techniques; *supervised learning* and *unsupervised learning.* Supervised learning is the learning process based on input-output pairs and correlate input to an output. In unsupervised learning, there is only input data and the learning goal is to identify the relationship between the variables and learn the structure of the data.


MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

The aim of supervised machine learning is to build a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data. Supervised learning uses classification and regression techniques to develop predictive models.

1. *Classification* techniques predict categorical responses, for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories.
2. *Regression* techniques predict continuous responses, for example, changes in temperature or fluctuations in power demand.

Problem Identification

In this study, the type of learning task is *supervised learning* as the input data is labeled and the output is supplied. The output is the polymer solution rheology, specifically the **viscosity** and it is a function of multiple variables including the shear rate, polymer concentration, salinity, type of electrolyte, temperature and molecular weight of the polymer solution. The output (viscosity) here is of continuous value which makes it a *regression* problem. The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value. The smaller the error the greater the accuracy of our regression model.

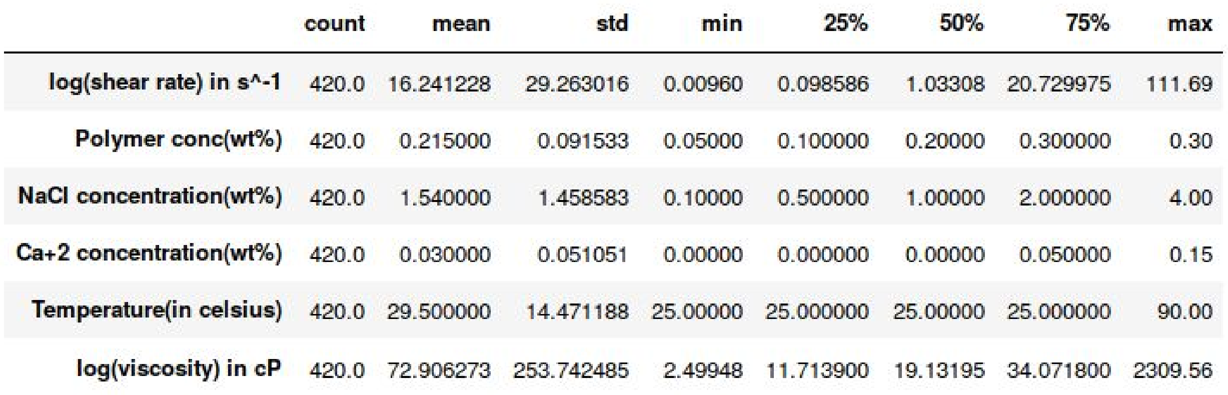
Understanding Data

Rheological data for an HPAM polymer (**Floppam 3330S**, molecular weight - 8 million Dalton) was compiled and studied. Dataset consists of **420 rows** and **6 columns** where each column represents ‘log(shear rate) in s^-1', 'Polymer conc(wt%)', 'NaCl concentration (wt%)', 'Ca+2 concentration(wt%)', 'Temperature(in celsius)' and 'log(viscosity) in cP’ respectively.

We performed many process/visualization to understand given data, some of them are listed below:-

Statistical Analysis

The description of each column of the dataset is given below.

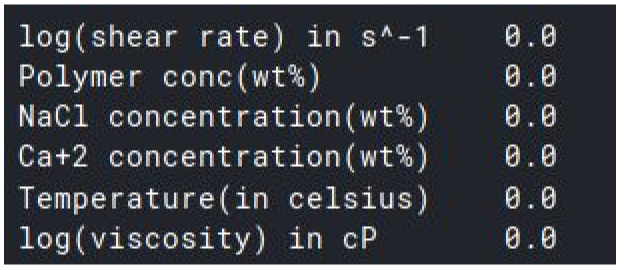


	count	mean	std	min	25%	50%	75%	max
log(shear rate) in s^-1	420.0	16.241228	29.263016	0.00960	0.098586	1.03308	20.729975	111.69
Polymer conc(wt%)	420.0	0.215000	0.091533	0.05000	0.100000	0.20000	0.300000	0.30
NaCl concentration(wt%)	420.0	1.540000	1.458583	0.10000	0.500000	1.00000	2.000000	4.00
Ca+2 concentration(wt%)	420.0	0.030000	0.051051	0.00000	0.000000	0.00000	0.050000	0.15
Temperature(in celsius)	420.0	29.500000	14.471188	25.00000	25.000000	25.00000	25.000000	90.00
log(viscosity) in cP	420.0	72.906273	253.742485	2.49948	11.713900	19.13195	34.071800	2309.56

We can easily see that our most data is **positively skewed**.
For viscosity data, the values for mean and median are following.
Mean: 72.906273 & Median: 19.13195
As the mean is greater than the median, it implies that data is positively skewed. Similarly, the polymer concentration, NaCl Conc., Ca+2 Conc., and Temperature shows some positive skewness.
Hence, the dataset before modeling was scaled using StandardScaler to normalise the data.

Missing Data (%):

The missing value percentage of all the columns in the dataset is given below:



```
log(shear rate) in s^-1     0.0
Polymer conc(wt%)           0.0
NaCl concentration(wt%)     0.0
Ca+2 concentration(wt%)     0.0
Temperature(in celsius)     0.0
log(viscosity) in cP        0.0
```

We can clearly see that the dataset contains no missing values and is completely cleaned, hence no preprocessing required.

Range of Properties:

Polymer	Polymer Conc. (wt%)	NaCl Conc. (wt%)	Ca+2 Conc. (wt%)	Temperature (C)
FP 3330S	0.05 - 0.30	0.1 - 4	0.0 - 0.15	25 - 90

Data Visualisation:

Data visualisation techniques such as scatter plot and correlation visualisation (heat map) were employed to understand the behaviour of the data.

Scatter Plot: Used for the single attribute to see its distribution over the entire dataset, so we can get the idea of skewness of given data. The fit regression line shows the effect of that attribute on the target variable(revenue), so we can know how each attribute depends on it. The scatter plot between shear rate and viscosity of the given polymer type is shown below: (figure S1 - linear plot, figure S2 - plot with polynomial of order=4)

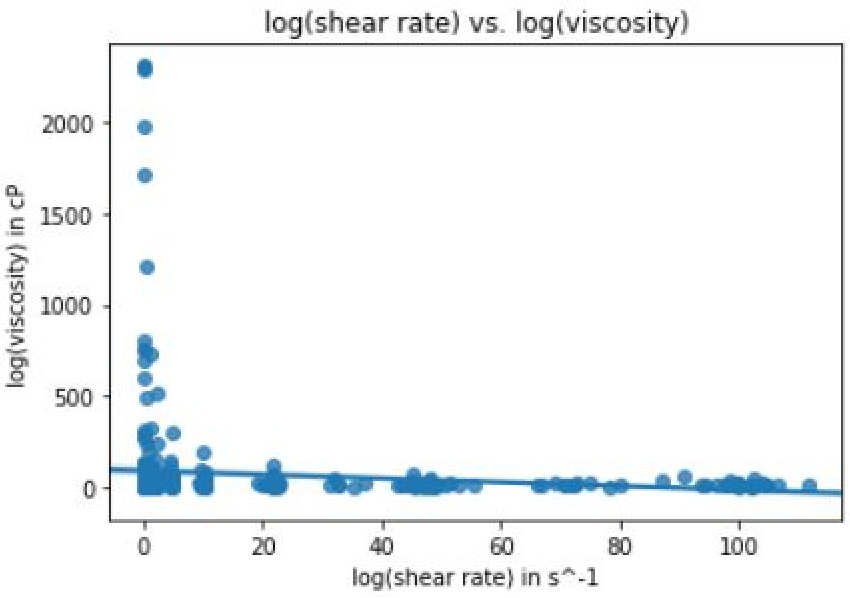


Figure S1

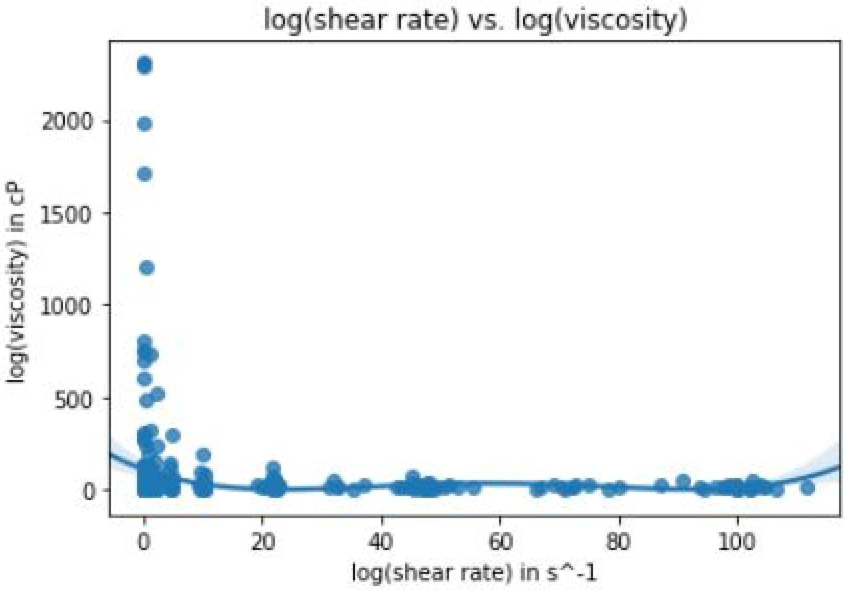


Figure S2

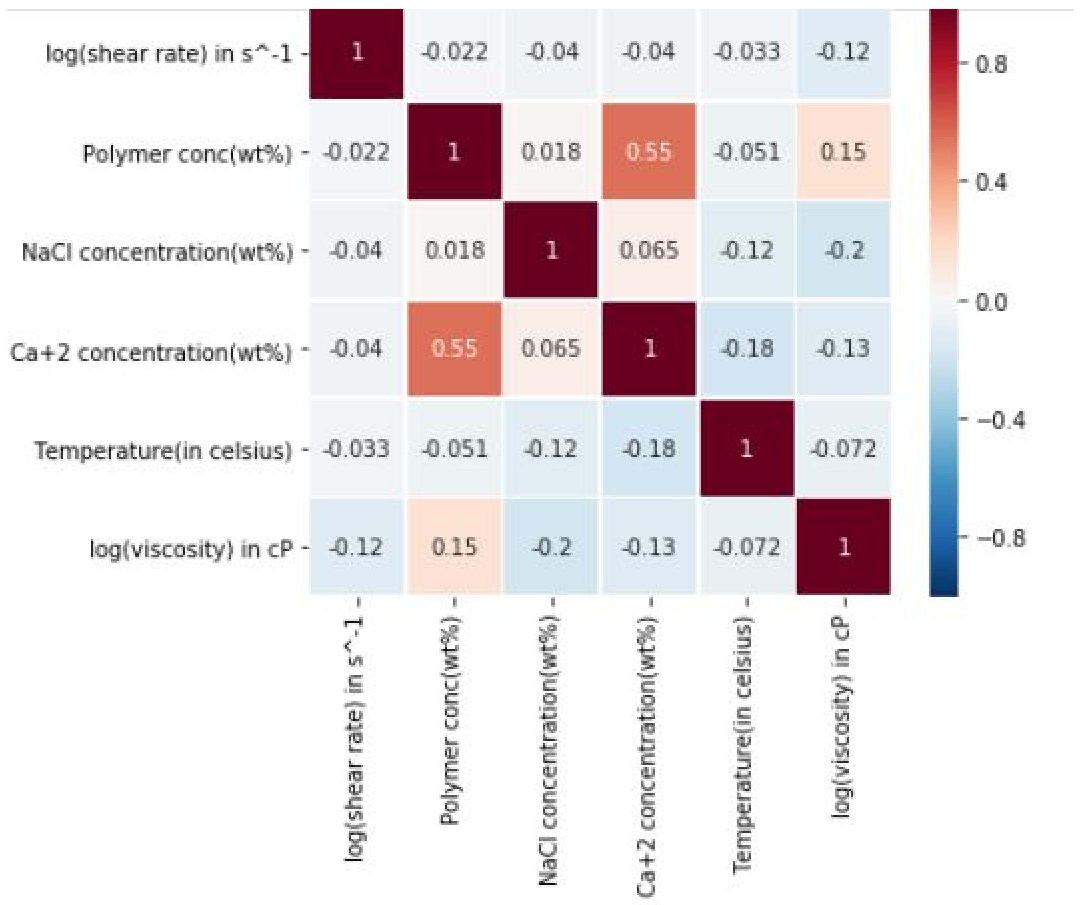
When we first plot our data on a scatter plot it already gives us a nice quick overview of our data. But it's also nice to be able to see how complicated our task might get; we can do that with regression plotting. In the figure S1 we've done a linear plot. The figure S2 uses a polynomial of order 4 and looks much more promising. So it looks like we'll definitely need something of at least order 4 to model this dataset. The model can be compared with their error value which is done in the next sections.

Correlation visualization (Heat Map):

Heat maps are 2D displays of the values in a data matrix. Larger values were represented by dark brown squares (pixels) and smaller values by lighter squares. It reflects the correlation between all these properties based on the dataset.

From the below heat map plot, following inferences can be drawn:

- The matrix value corresponding to polymer concentration and viscosity is **0.15**. This shows a positive correlation between them.
- The matrix value corresponding to salinity (NaCl Conc.) and viscosity is **-0.20**. This shows a negative correlation between them. This justifies the theoretical relation between salinity and viscosity. At low salinity, this feature is greatly emphasized when the negative charges on the backbone chain of the polymer repel each other, causing the flexible chain structure to elongate and, as a result, the viscosity increases.



Modeling Approaches

Training the system:

While training the model, data is usually split in the ratio of 80:20 i.e. 80% as training data and rest as testing data. In training data, we feed input as well as output for 80% data. The model learns from training data only. We use different machine learning algorithms (which we will discuss in detail in the next sections) to build our model. By learning, it means that the model will build some logic of its own.

Once the model is ready then it is good to be tested. At the time of testing, the input is fed from the remaining 20% data which the model has never seen before, the model will predict some value and we will compare it with actual output and calculate the accuracy.

The following machine learning models were used to train the dataset:

1. Multivariable Linear Regression
2. Non-Linear/Polynomial Regression
3. Artificial Neural Network (ANN)
4. Random Forest (RF)
5. Support Vector Machine (SVM)

1. Multivariable Linear Regression

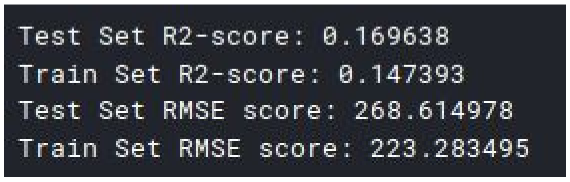
Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the linear relationship between the explanatory (independent) variables and response (dependent) variable. It can be formulated as:

$$y = B_1 * x_1 + B_2 * x_2 + \ldots + B_n * x_n$$

In this equation, the subscripts denote the different independent variables. x_1 is the value of the first independent variable, x_2 is the value of the second independent variable, and so on. It keeps going as more and more independent variables are added until the last independent variable, x_n, is added to the equation.

In this study, Linear Regression was used to predict viscosity based on polymer solution properties. The Linear Regression algorithm from the Scikit-Learn package, *LinearRegression*, was used to build the model.

R2 Score and RMSE value for training and test dataset were calculated as follows:



```
Test Set R2-score: 0.169638
Train Set R2-score: 0.147393
Test Set RMSE score: 268.614978
Train Set RMSE score: 223.283495
```

The comparison for actual and predicted value on y=x graph was done for both the training and test dataset.

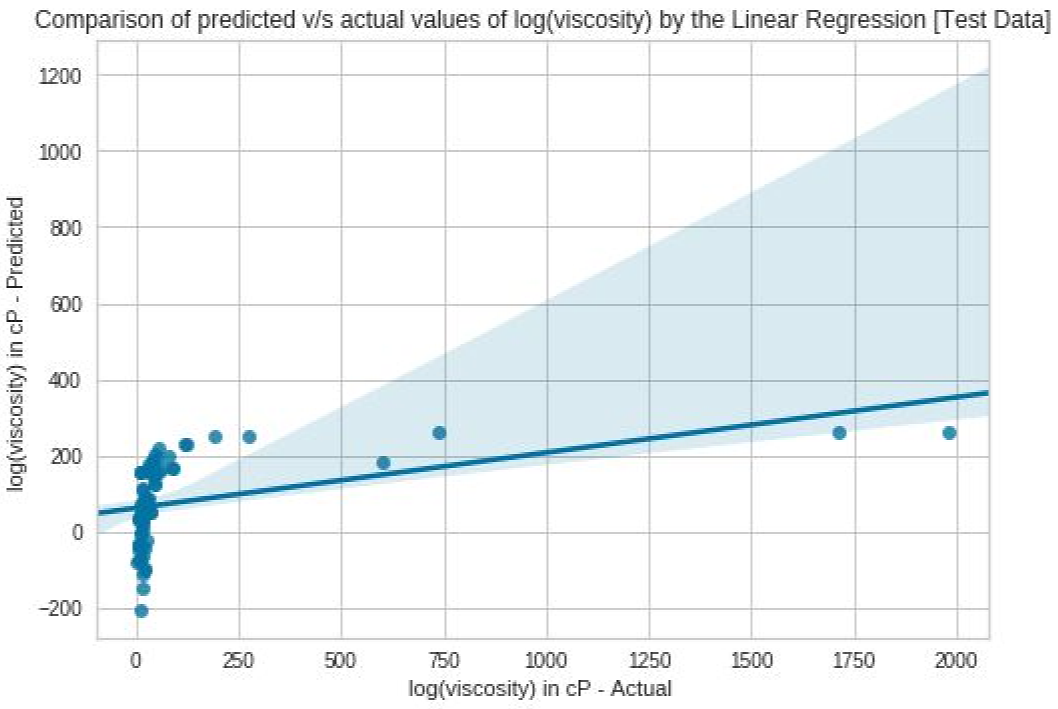


Figure M1(a) Comparison for predicted vs actual values - Test Data (Linear Regression)

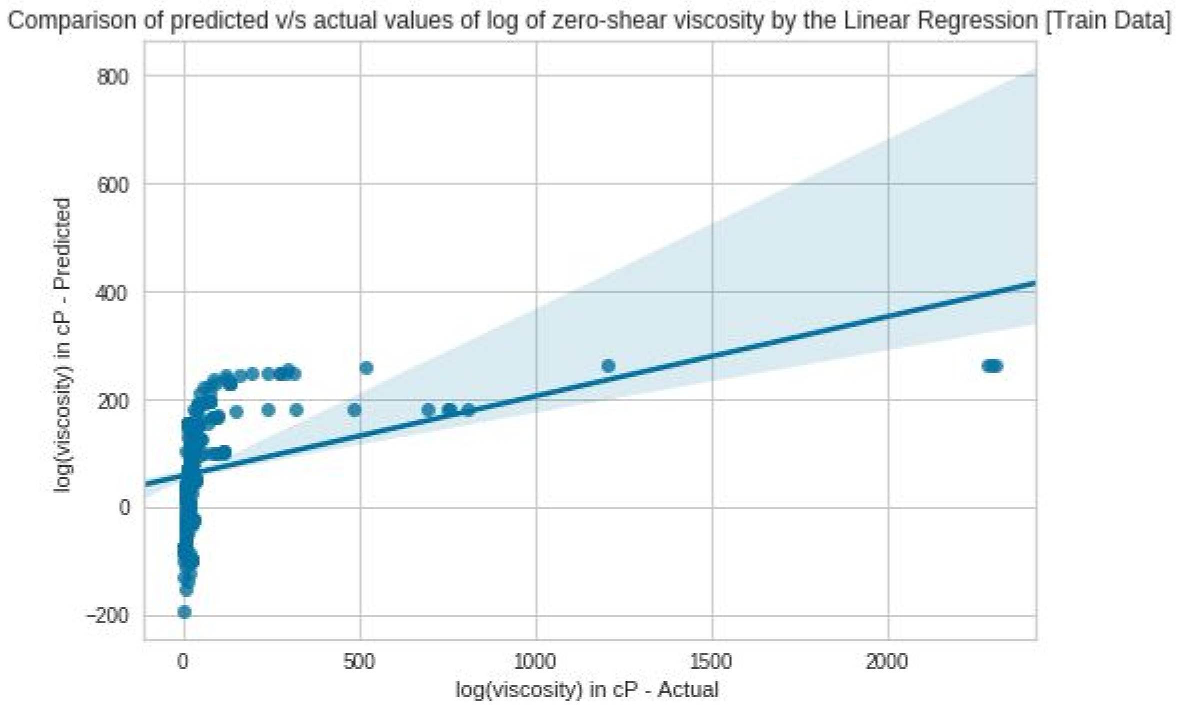


Figure M1(b) Comparison for predicted vs actual values - Train Data (Linear Regression)

Linear regression model seems to perform badly with the R2 score of 16.9% for the test dataset at modeling the prediction of viscosity. To improve the model, we try to include the non-linear modeling using Polynomial Regression.
Demerits of linear regression are:-

- Unable to learn complex relationships.
- Difficult to capture non-linear relationships (without first transforming data which can be complicated).

2. Polynomial Regression

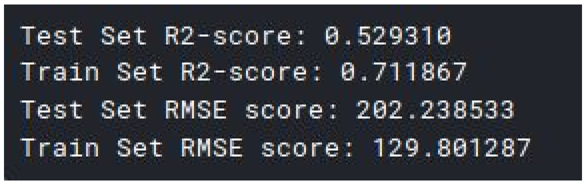
Polynomial Regression is a form of linear regression in which the relationship between the independent variables(x) and dependent variable y is modeled as an *nth* degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y|x).

Why Polynomial Regression:

- There are some relationships that can be hypothesized as curvilinear. Such types of cases will include a polynomial term.
- Inspection of residuals: If we try to fit a linear model to curved data, a scatter plot of residuals (Y axis) on the predictor (X axis) will have patches of many positive residuals in the middle. Hence in such a situation it is not appropriate.

In this study, Polynomial Regression was used to predict viscosity based on polymer solution properties. The Polynomial features were imported from the Scikit-Learn package, *PolynomialFeatures*, was used to build the model.

R2 Score and RMSE value for training and test dataset were calculated as follows:



```
Test Set R2-score: 0.529310
Train Set R2-score: 0.711867
Test Set RMSE score: 202.238533
Train Set RMSE score: 129.801287
```

The comparison for actual and predicted value on y=x graph was done for both the training and test dataset.

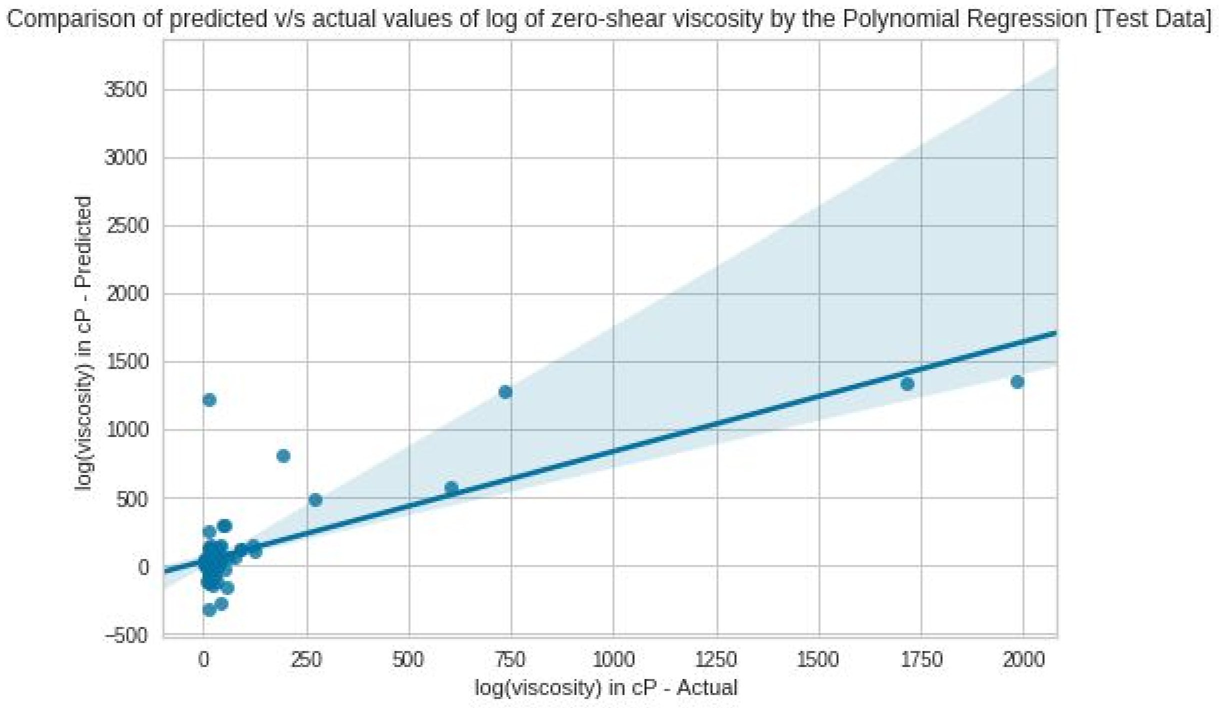


Figure M2(a) Comparison for predicted vs actual values - Test Data (Polynomial Regression)

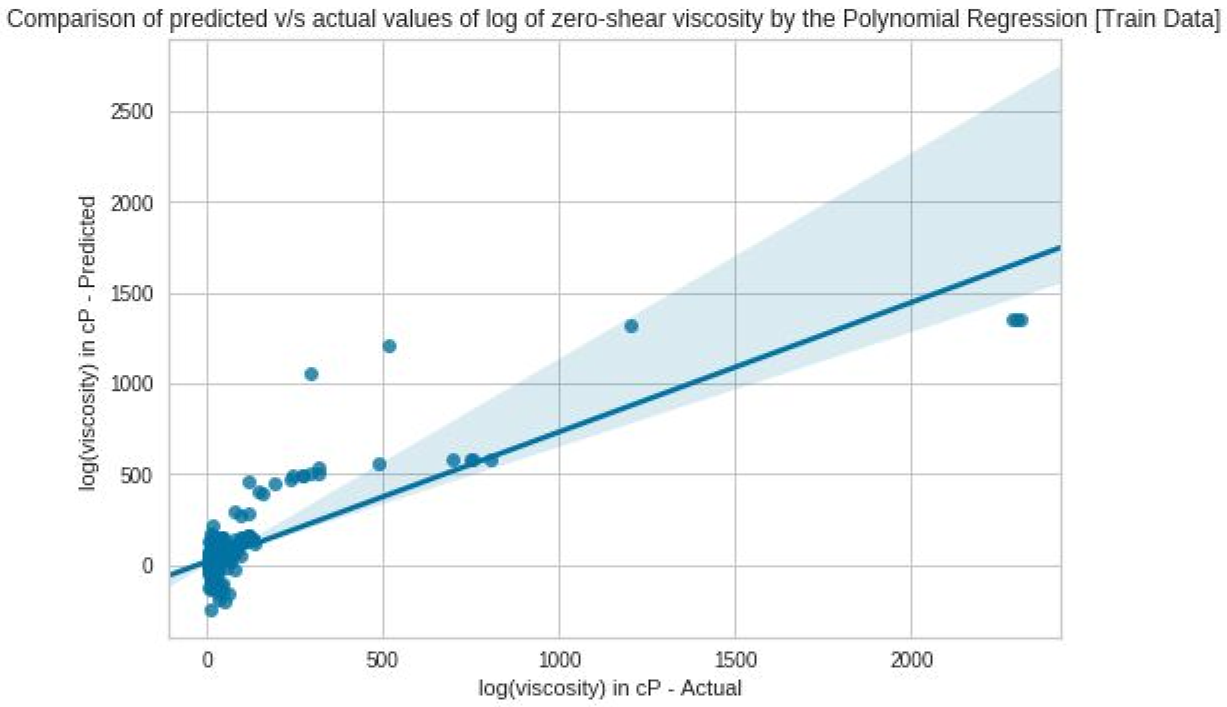


Figure M2(b) Comparison for predicted vs actual values - Train Data (Polynomial Regression)

The above plot shows that the polynomial regression give better results over linear regression.

3. Artificial Neural Network

Artificial Neural Networks (ANN), also called multilayer perceptron (MLP), are a sophisticated and powerful deep-learning model that consists of interconnected nodes (neurons) and layers that form a set of networks. It is called "neural" because it was inspired by neuroscience and the biological neurons that constitute animal brains. Figure M1 shows a typical configuration of an artificial neural network. It comprises three main elements: input layer, hidden layer(s), and output layer. The input layer holds a number of neurons corresponding to the number of features of the problem. The output layer contains neurons equal to the number of expected outputs. For our problem,
Input layer: 5 Neurons (shear rate, polymer concentration, salinity, hardness, temperature)
Output layer: 1 Neuron (Viscosity)

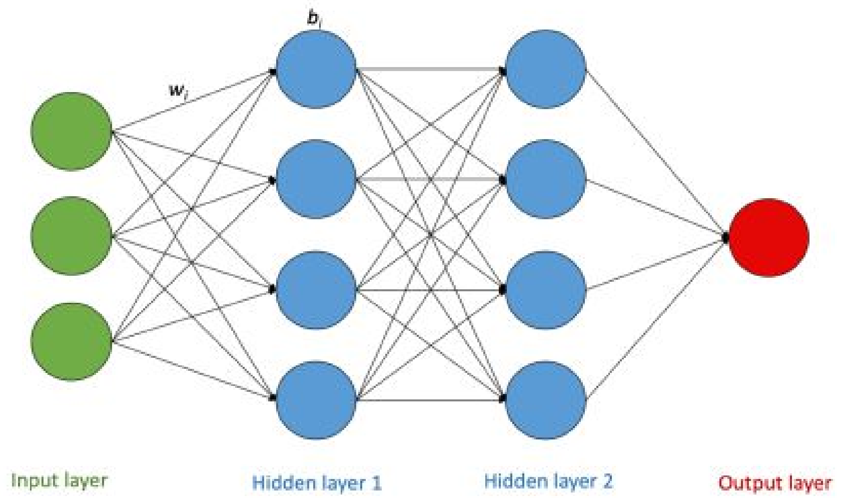


Figure M1

The hidden layers are the most critical element of ANN because that is where the calculation occurs. The hidden layers may have as many layers as the problem dictates, but they must have at least one layer. There is no exact rule about how many neurons should be in each hidden layer because that quantity depends on problem-related factors such as numbers of features and outputs. However, there are some practice-based suggestions to start with and perhaps modify as necessary.

The value of the neuron is communicated from the neurons in the preceding layer through the following equation:

$$z = \sum_{i=1}^{m} w_i x_i + b$$

where **z** is the current neuron value, **w**$_i$ is the connection weight between the current and previous neurons, **x**$_i$ is the value of the preceding neuron, **b** is the bias of the current neuron, and **m** is the number of neurons in the previous layer.

Furthermore, activation functions are used at the end of a hidden neuron to introduce ***nonlinear complexities*** to the model. Their main purpose is to convert an input signal of a neuron to an output signal. That output signal now is used as an input in the next layer. Thus, when the value of z is calculated, it is further used in the activation function. The figure M2 depicts the same about activation function.

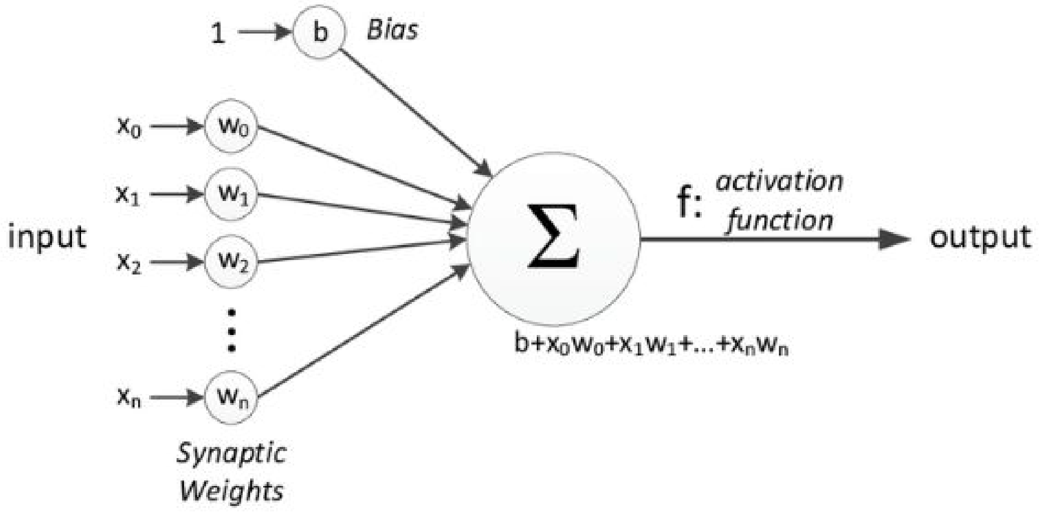
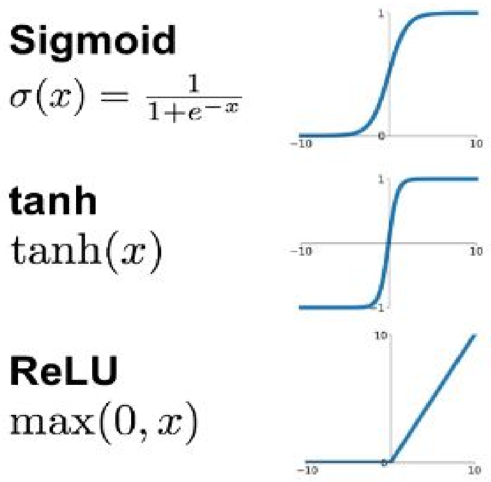


Figure M2

The most common types of activation functions are:



Note: A neural network without activation function would simply be a linear regression model, which is limited in its complexity and has less power to learn complex functional mappings from data.
In this study, the ANN model was trained with the train dataset (80% of the total available data) used in the previous models. A feed-forward artificial neural network with a back-propagation algorithm, "**MLPRegressor**" from the Scikit-Learn library, was used in this study. The construction of the model was done by a series of trial-and-error attempts in order to produce the best accuracy.

During the construction process, three main features of ANN were studied: *the activation function*, *the solver*, and *the number of neurons in each hidden layer.*

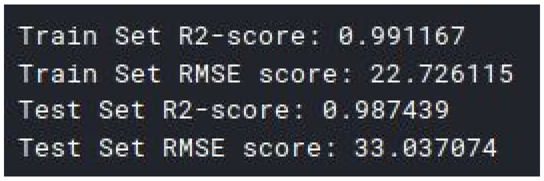
The three activation functions mentioned earlier were examined. Each function was applied and its performance compared with the others. The second important feature is the solver function, which is for the weight optimization. The MLPRegressor offers three types of solver algorithms: 'LBFGS', 'SGD', 'Adam'. Similar to the activation function, all three solvers were studied. Following results were obtained:

Solver	**Activation Function**	**Test Set Accuracy (%)**
lbfgs	logistic/sigmoid	82.92
	tanh	73.02
	ReLu	98.74
SGD	logistic/sigmoid	44.60
	tanh	7.10
	ReLu	-6.75

adam	logistic/sigmoid	-7.52
	tanh	-7.22
	ReLu	14.4

*As a result, **'ReLU'** was found to yield the most accurate predictions with the **'lbfgs'** solver.*

Finally, the number of hidden layers was examined. Sometimes, an exceedingly large number of hidden layers may result in an overfitting scenario, and vice versa. Thus, the models were initially built with one hidden layer, and the results were investigated. After several trials, the optimal number of hidden layers used in the ANN model was found to be two. (Each with a number of neurons = 7).
The following evaluation metrics measure were taken:



```
Train Set R2-score: 0.991167
Train Set RMSE score: 22.726115
Test Set R2-score: 0.987439
Test Set RMSE score: 33.037074
```

The RMSE score of the train set and test set are close enough which gives a signal for good modeling. The comparison for actual and predicted value on y=x graph was done for both the training and test dataset. Both the training and the testing sets show an excellent match with the actual data.

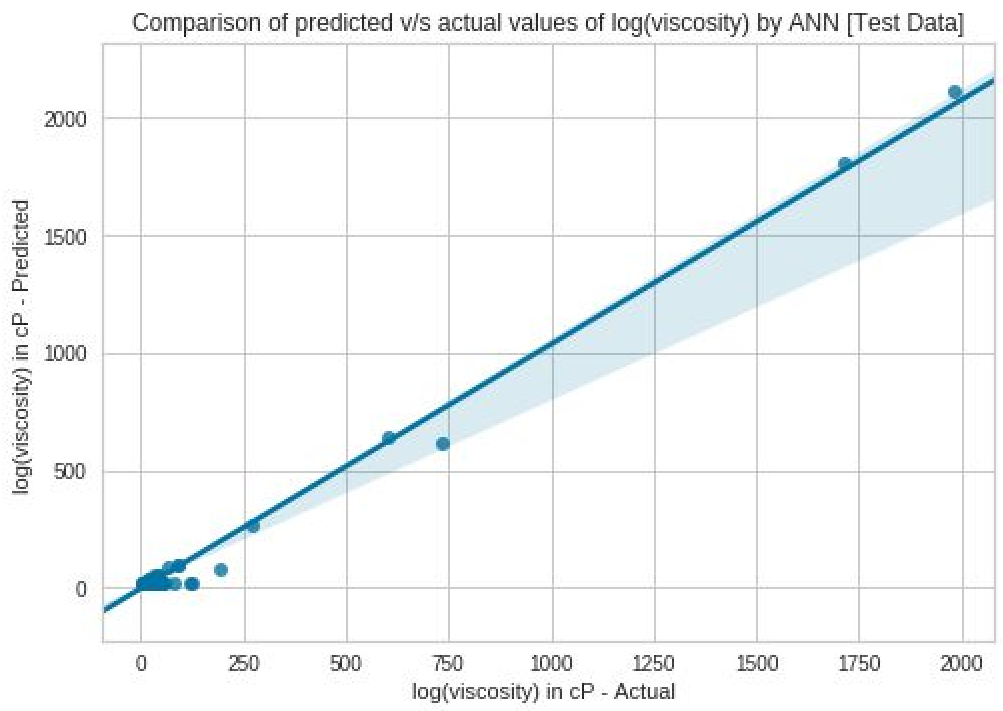


Figure M3(a) Comparison for predicted vs actual values - Test Data (ANN)

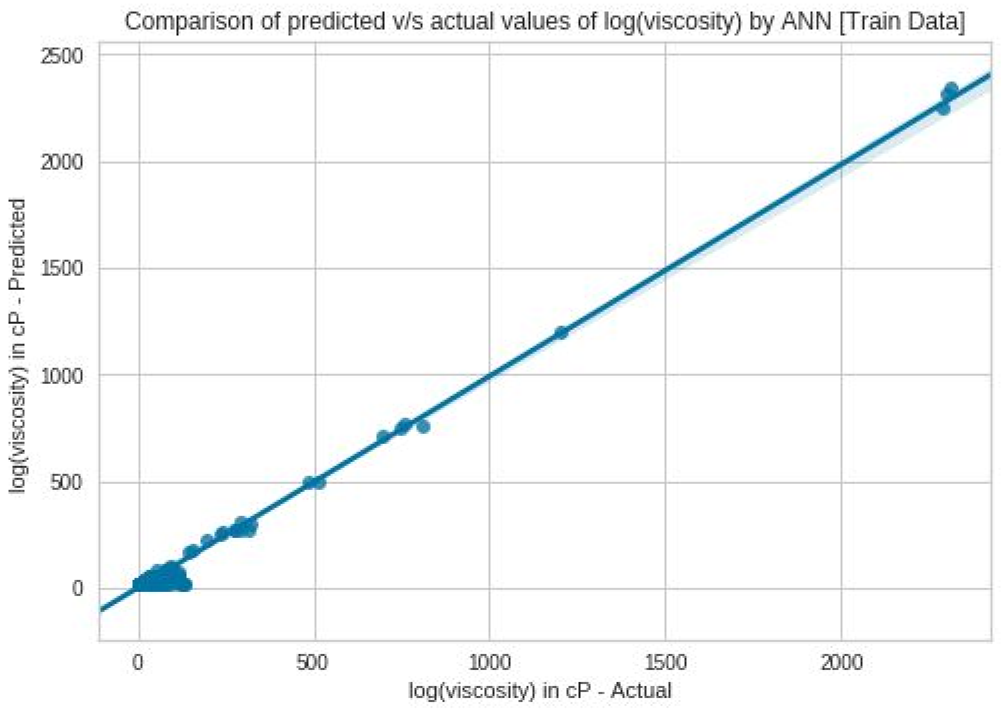


Figure M3(b) Comparison for predicted vs actual values - Train Data (ANN)

4. Random Forest

Random Forests is a tree-based machine learning method that can be applied to regression and classification problems. It is a collection of multiple decision trees "forest" that are trained simultaneously, and the output is the mean prediction of each individual tree. It has gained much popularity in the last few years due to its robust predictive capability.

Using the concept of decision trees, it is easier to comprehend the idea of Random Forests. As previously stated, Random Forests algorithm is a set of decision trees that are trained in a completely independent manner. This means that the splitting rules are completely independent between the trees, and those rules are drawn randomly. Also, each tree is trained by a different dataset, called "bootstrap samples," taken from the same training dataset, which adds additional randomness. The major advantage of the randomness in RF is the significant reduction in variance without affecting the bias. Furthermore, another important parameter of RF is the optimal number of trees, which is technically case-dependent. Normally, trees are continually added until no further improvement is seen in terms of error reduction. Once the RF is constructed and the trees are trained, the output value is the mean of multiple values produced from each tree.

Working of RF Algorithm:

The following steps elaborate on the working of Random Forest Algorithm –

- First, it starts with the selection of random samples from a given dataset.
- Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- In the third step, voting will be performed for every predicted result.
- At last, the most voted prediction result is selected as the final prediction result.

The following diagram will illustrate its working –

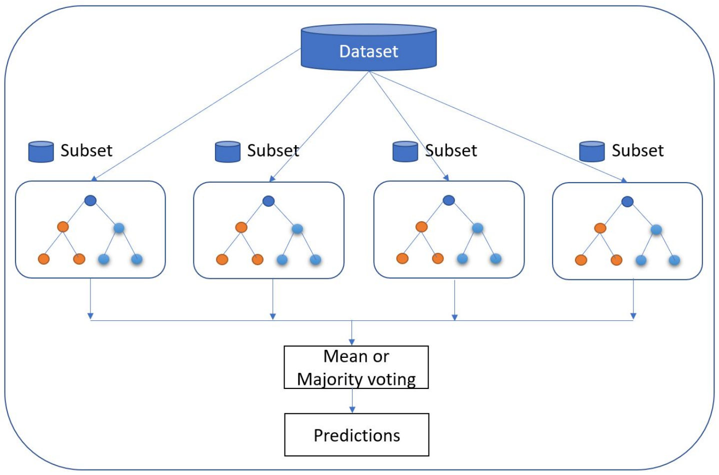


Figure M4 (Working of RF Model)

In this study, RF was used for the same objective as the regression approach: to predict viscosity based on polymer solution properties. The RF algorithm from the Scikit-Learn package, Random Forests Regressor, was used to build the RF model.

Figure M5 shows an example of one tree from an RF model. The depth of trees was limited to 3 in the forest to produce an understandable image. At each node, i.e. rectangle, the splitting feature and its condition are shown at the top. Then, following next is the mean squares error (MSE), which is the mean of the error squared between the actual viscosity data of the samples that fall in that node and the node's average viscosity value. Clearly, the nodes at the early stages of the tree tend to have higher MSE. Therefore, further splitting was applied until the minimum number of samples per node was reached, which is set at two samples.

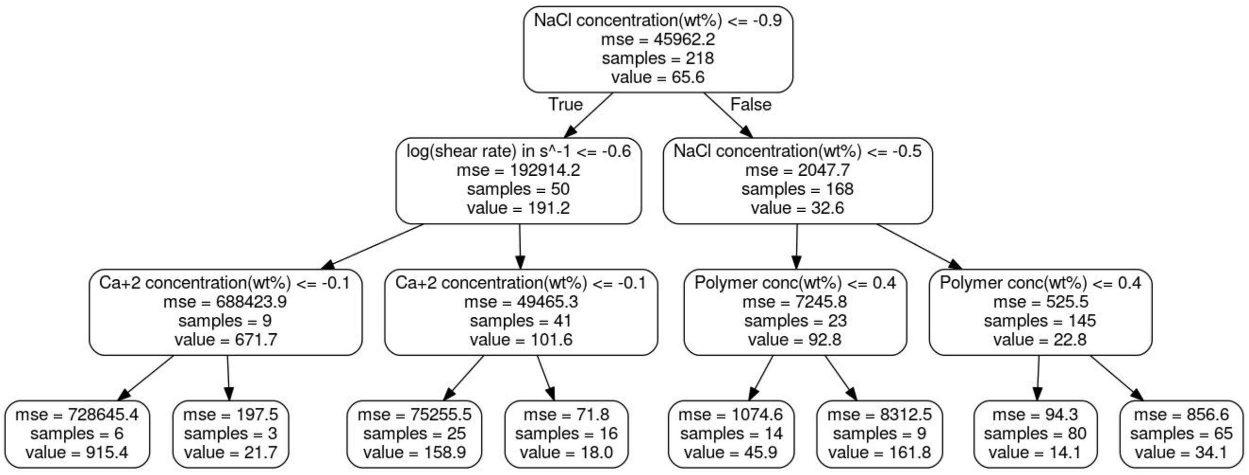
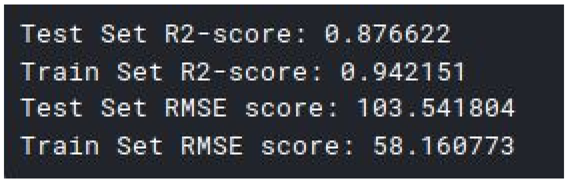


Figure M5 (Decision Tree Example drawn from an RF model)

The following evaluation metrics measure were taken:



```
Test Set R2-score: 0.876622
Train Set R2-score: 0.942151
Test Set RMSE score: 103.541804
Train Set RMSE score: 58.160773
```

Predictions made by the Random Forest model outperformed the predictions by both Linear and Non-Linear Regression models. The comparison for actual and predicted value on y=x graph was done for both the training and test dataset.

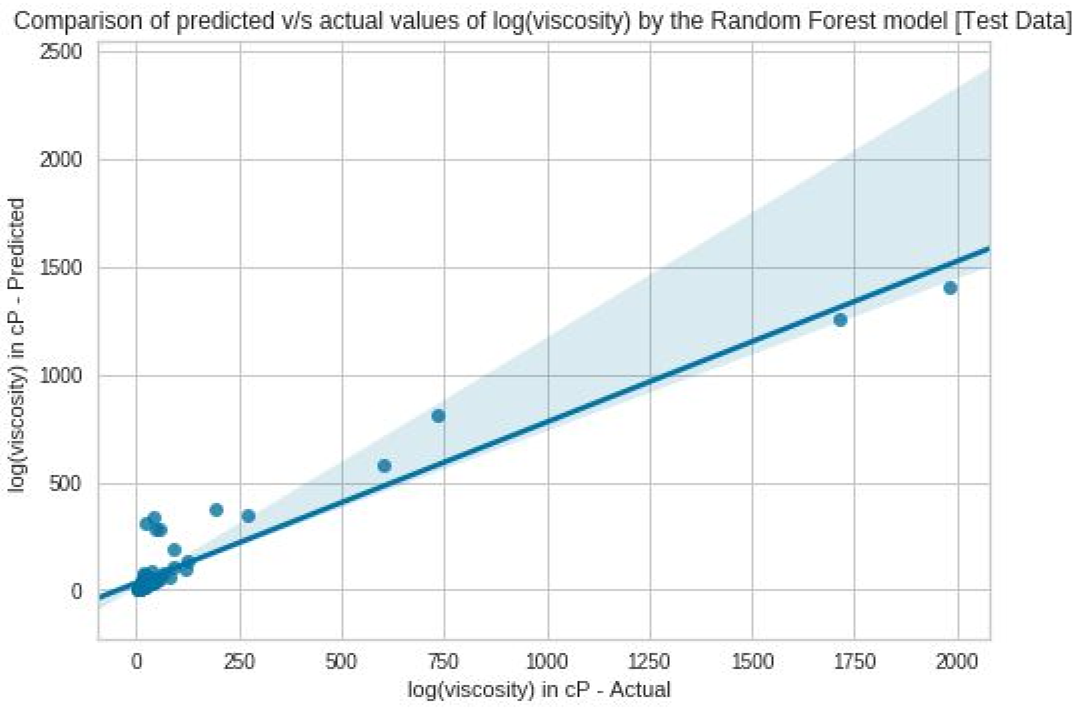


Figure M6(a) Comparison for predicted vs actual values - Test Data (RF)

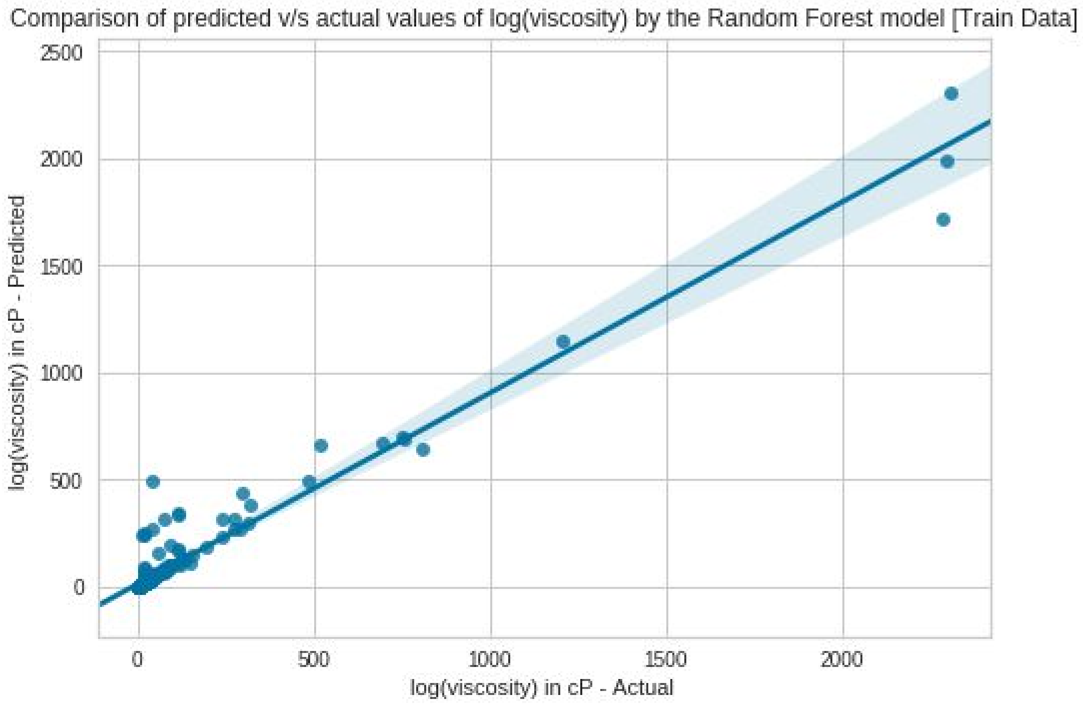
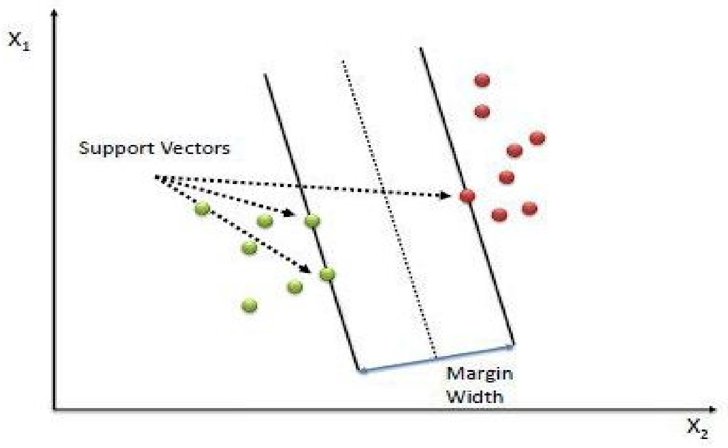


Figure M6(b) Comparison for predicted vs actual values - Train Data (RF)

5. Support Vector Machine (SVM)

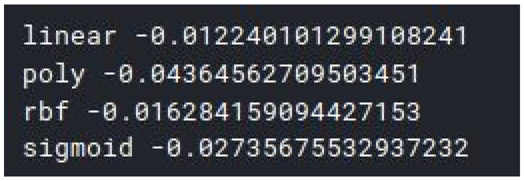
Support vector machines so called as SVM is a ***supervised learning algorithm*** which can be used for classification and regression problems as support vector classification (SVC) and support vector regression (SVR). It is used for smaller dataset as it takes too long to process. SVM is based on the idea of finding a hyperplane that best separates the features into different domains.
The points closest to the hyperplane are called as the ***support vector points*** and the distance of the vectors from the hyperplane are called the ***margins***.



An SVM is a kind of **large-margin classifier:** it is a vector space based machine learning method where the goal is to find a **decision boundary** between two classes that is maximally far from any point in the training data. While other linear regression models try to minimize the error between the predicted and the actual value, Support Vector Regression tries to fit the best line within a predefined or threshold error value.

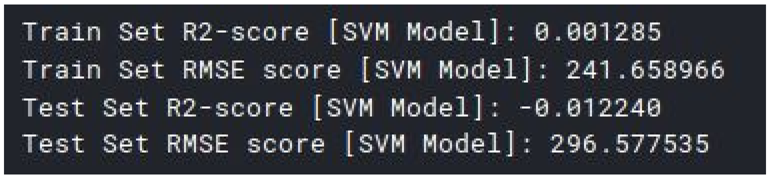
In this study, the SVM model was used for the same objective as the regression approach: to predict viscosity based on polymer solution properties. The SVM algorithm from the Scikit-Learn package, Support Vector Regressor (SVR), was used to build the SVM model.

There is a parameter to svm.SVR for example which is kernel. Think of a kernel like a transformation against your data. It's a way to grossly simplify your data. This makes processing go much faster. In the case of svm.SVR, the default is rbf, which is a type of kernel. The other choices are 'linear', 'poly', 'sigmoid'.
The R2 scores for different kernels were computed and found as below:



```
linear -0.012240101299108241
poly -0.04364562709503451
rbf -0.016284159094427153
sigmoid -0.02735675532937232
```

None of the models seems to give any substantial performance over predicting the viscosity. The evaluation metric for SVM model with linear kernel were reported as below:



```
Train Set R2-score [SVM Model]: 0.001285
Train Set RMSE score [SVM Model]: 241.658966
Test Set R2-score [SVM Model]: -0.012240
Test Set RMSE score [SVM Model]: 296.577535
```

The comparison for actual and predicted value on y=x graph was done for both the training and
test dataset.

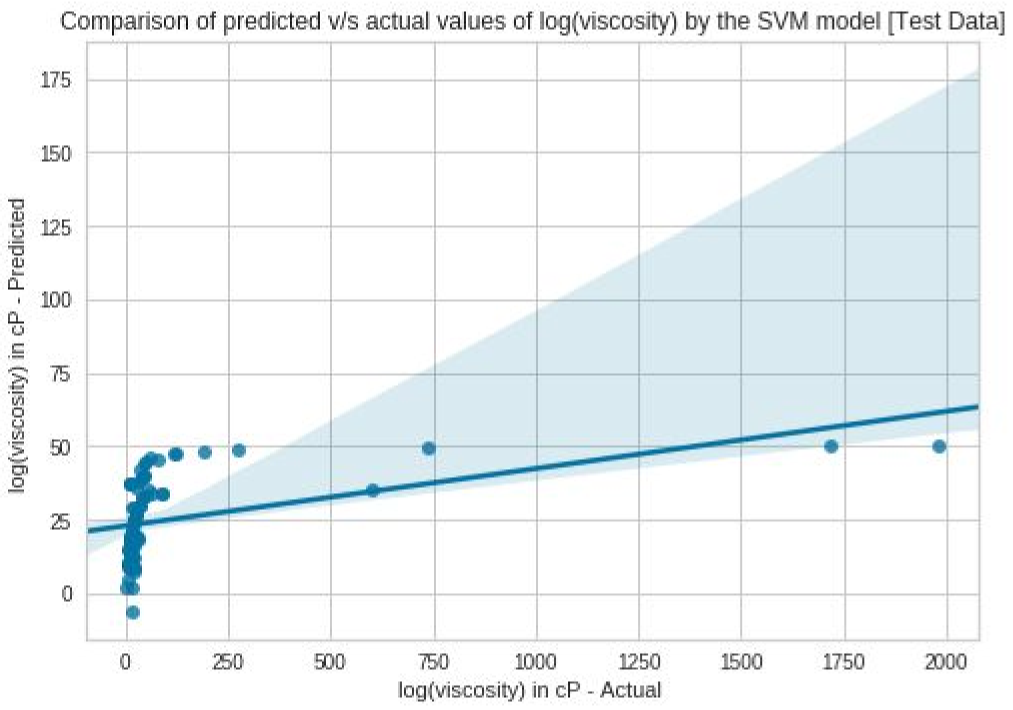


Figure M7(a) Comparison for predicted vs actual values - Test Data (SVM)

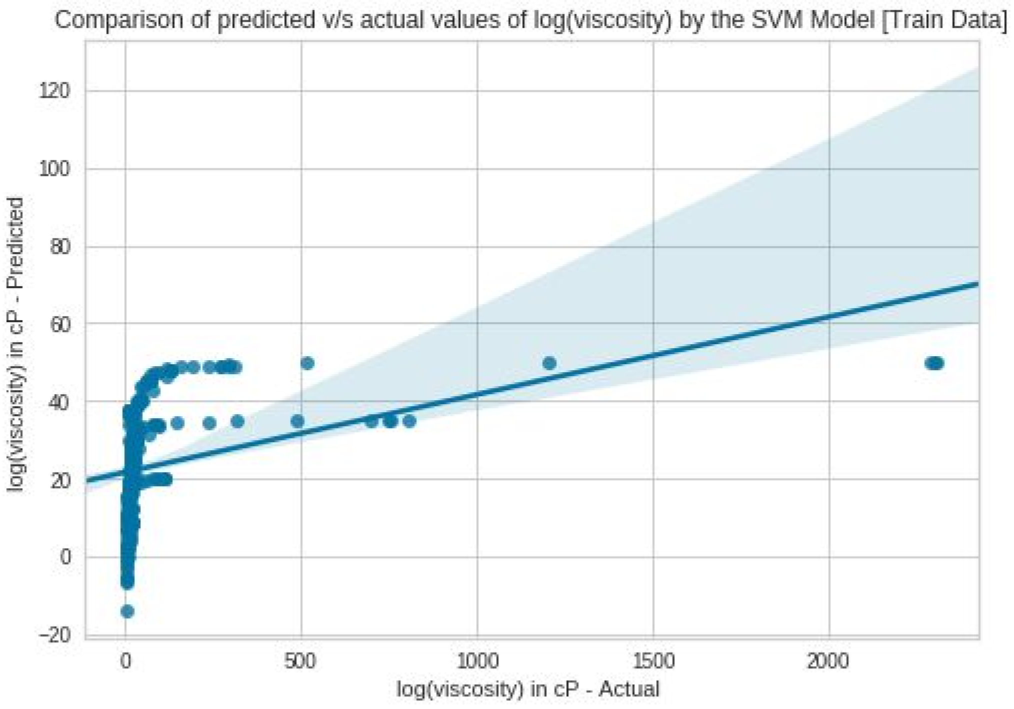


Figure M7(a) Comparison for predicted vs actual values - Train Data (SVM)

Comparison & Results

Now, we compare the performance of all the models applied so far with their errors (RMSE score) and R2 score over the test dataset.

Algorithm	R2 Score	RMSE Value
Multivariable Linear Regression	16.96 %	268.61
Polynomial Regression	52.93 %	202.83
Artificial Neural Network	98.74 %	33.03
Random Forest	87.66 %	103.54
Support Vector Machine	1.22 %	296.57

Our most useful algorithms are Neural Network and Random Forest which performed very well with the test dataset. ANN models seem to outperform RF, SVM and regression models. And, the worst performed model is Support Vector Machine (SVM) which gave a very poor R2 Score and high RMSE value.

Conclusion and Suggestions

The major emphasis was given to learn and understand the results obtained by the ANN and RF model and compared with the results specified in the given thesis.

The results described earlier indicate that the modeling approaches used in this study have proved to be reliable and consistent. These results have demonstrated the models' capability of simulating the rheological properties while minimizing the uncertainty. For instance, by looking at ANN model predictions, one can realize its useful impact on rheological studies, especially ones that involve chemical EOR applications. Furthermore, this study revealed the powerfulness of some advanced machine learning models, i.e., RF and ANN. The result of combining EOR and ML sciences led to the development of more reliable and accurate predictive tools that can add tremendous value to relevant research studies and applications. Products of this combination can be further applied to similar problems or perhaps can replace existing models or correlations.

The models presented in this study can be very useful in many EOR-related applications. For instance, a quick and easy way to calculate a polymer solution's viscosity is by knowing its concentration and salinity. However, an important application is implementation into a reservoir simulator (e.g.

UTCHEM). The proposed models can enhance the accuracy of the polymer solution viscosity calculated in the simulator. Also, it may significantly reduce the uncertainty and improve the simulator's reliability.

Some recommended future work includes the following goals:

- To improve the existing models by compiling additional rheology data, either by conducting rheology experiments or by collecting data from literature and feeding it into these models.
- The accuracy score for the Random Forest model can be improved by hyperparameter optimisation.
- Implement the new models to a chemical flood reservoir simulator. The models can be ANN which is the most efficient, beside its very good accuracy.

THANK YOU