Workflow for Predicting Undersaturated Oil Viscosity using Machine Learning

Sofianos Panagiotis Fotias [1], Vassilis Gaganis [1,2,*]

May 30, 2023

[1]Mining and Metallurgical Engineering, National Technical University of Athens, Athens 157 73, Greece
[2]Institute of Geoenergy, Foundation for Research and Technology, Chania, 73100, Greece
Correspondence: sfotias@metal.ntua.gr

Abstract

Undersaturated oil viscosity is a dominant fluid parameter to be measured in oil reservoirs due to its direct involvement in flow calculations. Since PVT experimental work is expensive and time costly, prediction methods are essential. This work presents the utilization of viscosity data from more than five hundred fluid reports with the purpose of developing data driven models to predict undersaturated oil viscosity using easy-to-get measurements. The suitability of popular machine learning techniques in performing this task is also examined by comparing the models obtained for each method using several popular statistical metrics. A complete workflow for this process is introduced to demonstrate the integrity of the process followed and to guide in further research in predicting similar PVT properties.The workflow showcases the advantages of combining engineers expertise to the art of data driven models developement, specifically on accuracy and ease of implementation, as well as their limitations.

Keywords: undersaturated oil viscosity; correlations; predictive methods; machine learning; supervised regression;

1 Introduction

Undersaturated oil viscosity (μ_o) refers to the viscosity of live oil at pressures above the bubble point (P_b), which typically correspond to the pressure range that prevails during most of a reservoir's lifespan. Although oil viscosity varies only with pressure in undersaturated reservoirs with a constant composition, the degree of this variation may be quite drastic. Indeed, the extent of viscosity change can range from 0.5% to 40% per 1,000 psi, in contrast to volumetric properties such as the oil formation volume factor (B_o) which typically varies from 0.5% to 2.8% per 1,000 psi. This indicates that a reduction in pressure during production has a significant and direct impact on viscosity, which in turn could affect the performance of the reservoir as the latter is directly involved in the flow laws.

The exact measurement or even a fair computational estimate of viscosity is essential in all flow calculations in reservoir and production engineering [1], as it directly affects the pressure drop associated with flow. For example, Darcy's equation for steady state cylindrical (radial) flow in the near-wellbore area incorporates viscosity in the mobility ratio as shown in eq. (1) [2].

$$P_{\mathrm{e}} - P_{wf} = \frac{q_o \mu_o B_o \left[\ln\left(\frac{r_e}{r_w}\right) + S\right]}{7.0815 \cdot 10^{-3} kh} \tag{1}$$

where $P_e - P_{wf}$ is the pressure drawdown in the reservoir, k/μ_o is the mobility ratio, r_e/r_w is the geometry factor and S is the skin factor. The analysis of fluid flow in pipelines, such as the production tubing and the surface network, involves the application of the continuity equation and of the momentum conservation, for the prediction of pressures and flow rates. [3]. In that case, viscosity is incorporated in the pressure drop as shown by:

$$\frac{\Delta P}{\Delta L} = \frac{1}{144}\left[\rho_o \cos(\theta) + \frac{f\rho_o\mu_o^2}{2gd} + \frac{\rho_o\mu_o\Delta\mu_o}{g\Delta L}\right]$$

where ρ_o is the oil density, θ is the pipe inclination and d is the pipe diameter. The three parts of the equation's right hand side correspond to the hydrostatic, frictional and kinetic energy losses in the system, respectively. The frictional factor f, indirectly involves viscosity as it is obtained from the Moody Friction factor chart which is a function of the Reynolds Number [4] with the latter being calculated by

$$R = \frac{1488 \cdot d \cdot \nu \cdot \rho_o}{\mu_o}$$

where ν is the fluid's velocity.

Introducing accurate oil viscosity values to a flow simulator is of utmost importance as all errors in the fluid properties can only be compensated by means of history matching. Depending on the error in the viscosity value, the fluids might be considered to flow easier or more difficult than what really happens. History matching would try to account for that by abnormally modifying complementary terms such as the relative permeability of the oil phase (e.g. by modifying the parameters of a Stones model). This way the reservoir model can be twisted, thus reducing its prediction capability for the future production scenarios that need to be evaluated.

Dead oil viscosity is typically determined in the PVT (pressure-volume-temperature) laboratory through a standardized procedure. The dead oil sample is prepared in a controlled temperature environment and then introduced into the rotational viscometer. The rotational speed is increased gradually and the torque necessary to maintain the rotation is measured and used to calculate the viscosity of the oil using the viscometer's calibration data. On the other hand, measuring live oil viscosity can be time-consuming and expensive as it requires apparatus that utilizes the measurement of the time required for a rolling nickel ball, affected by shear and pressure of the fluid, to travel a pre-determined distance at controlled conditions. The ball is positioned inside the measuring barrel with the test fluid sample so that it is limited to an only rolling type motion and an electronic timer records the time required for the ball to roll through the barrel. This procedure requires skilled personnel, as the oil sample is under high pressure and temperature. When detailed lab values are not available, correlations can be used instead to estimate the reservoir oil viscosity [5]. These correlations typically utilize pressure and bubble point viscosity (μ_{ob}) as the most significant input parameters, while other properties such as fluid GOR (Gas Oil Ratio), API gravity and dead oil viscosity (μ_{od}) are considered secondary and used by few correlations only.

Developing a correlation typically involves collecting a number of physical observations and generating a function that captures the dependency between these observations. When physical evidence is utilized to select the appropriate function form, the correlations are referred to as physics-driven models. In contrast, Machine Learning (ML) methods are used to uncover patterns and relationships in the data that may not be apparent to the researcher. It is important to note, however, that scientist's experience and intuition are still critical in handling the data, choosing the correlating function form and in developing a ML model that performs optimally.

The utilization of ML methods for the prediction of PVT properties is an actively researched area. Various methods have been explored for the prediction of bubble point pressure [6], formation volume factor [7], dead oil viscosity [8] and even multiple PVT properties ensembles [9], [10], [11]. When it comes to undersaturated oil viscosity however, ML research has been limited. Notable attempts include data-driven correlations that have been generated with symbolic regression [12] which is an approach based on genetic algorithms [13] and ensemble methods with support vector machines [14]. In this work, several ML algorithms have been utilized for the prediction of undersaturated oil viscosity, incorporating a dataset that consists of more than 500 PVT reports. A complete, easy-to-implement

workflow is followed and presented. Furthermore, ML models generated following this workflow are compared to each other and results are drawn.

The rest of the paper is organized as follows. Correlation-based models are briefly explored in Section 2, to showcase the strenuous and time consuming process that researchers went through for their development. Section 3 details the characteristics of the dataset utilized as well as the quality control procedure that it went through. Section 4 provides a comprehensive account of the workflow employed for optimizing the models, including a description of the algorithms utilized. Subsequently, Section 5 presents the results obtained from the study followed by a broad discussion on the results and the outline of the process limitations. Finally, conclusions and suggestions are drawn in Section 6 based on the findings of this work.

2 Correlation review

In this section, the correlations commonly available to the industry to predict undersaturated oil viscosity are presented. The selection of these correlations is based on their historic significance, popularity and integration into commercial software packages for handling reservoir or pipeline flow problems. Rather than the pressure (P) itself, the available methods utilize either the pressure ratio or the pressure differential (i.e., P/P_b and $P - P_b$ respectively) as the primary correlating parameter, along with bubble point viscosity (μ_{ob}). Additional PVT properties, such as solution GOR, API gravity and dead oil viscosity (μ_{od}), are used by some correlations only. The most pronounced correlations can be found in any commercial software running flow calculations [15].

It is important to point out that when additional viscosity values in the undersaturated pressure range are known apart from μ_{ob}, a technique that involves modifying known correlations by applying shifting and multiplying coefficients may be used. By using this "tuning" approach, the original viscosity prediction is replaced by $\mu'_o = \alpha\mu_o + \beta$ which matches optimally the extra viscosity measurements and is now adequate in predicting the value of undersaturated oil viscosity accurately. Software packages like IPM [15] already utilize this method to improve accuracy in their predictive models.

Beal (1946) [16] correlated graphically 52 viscosity measurements from California and noted a viscosity increase with pressure which was greater with an increasing bubble point viscosity. Later, Standing (1977) [17] generated correlation equations for Beal's graphical method, resulting in a model linear in $\Delta P = P - P_b$ and a slope that depends polynomially on μ_{ob}. Kouzel (1965) [18] correlated data points with an exponential model known as the Barus' model $\mu = \mu_{ob}e^{\alpha(P-P_b)}$, while Vazquez and Beggs (1976) [19] were first to gather thousands of measurements across the world and generate a model exponental in the pressure ratio P/P_b with a pressure dependant exponent. Labedi (1982) [20] created two models, one for Libyan reservoirs and one for Nigerian and Angolan ones. Both models are linear in the pressure ratio and they utilize a polynomial relationship on dead oil viscosity, bubblepoint pressure and an exponential one on API gravity. Khan (1987) [21] correlated measurements from Saudi Arabian reservoirs using Barus' model, whereas Petrosky (1990) [22] gathered data points from the Gulf of Mexico to generate a model similar to Beal's with the exception of the slope depending exponentially on μ_{ob}. Kartoatmodjo and Schmidt (1991) [23] utilized thousands of PVT data points from North and South America, Southeast Asia and the Middle East to modify the parameters in Beal's model.

Al-Khafaji, Abdul-Majeed and Hassoon (1987) [24] used Middle East samples to generate a model that is polynomial in ΔP and exponential on API gravity, while Abdul-Majeed, Kattan and Salman (1990) [25] built on the previous model by additionaly utilizing an exponential relationship on GOR. Orbey and Sandler (1993) [26] correlated a model using Barus' equation, however, unlike previous researchers, they correlated different exponent values for paraffinic, naphthenic and aromatic hydrocarbons. De Ghetto, Paone and Villa (1994) [27] evaluated thousands of measurements, by splitting their data into four classes, based on API gravity and correlating equations, and modified existing models that performed best. Almehaideb (1997) [28] generated a model similar to that of Vazquez and Beggs on data points from UAE reservoirs and claimed an imporved performance due to the inclusion of GOR. Elsharkawy and Alikhan (1999) [29] developed models based on Middle Eastern data points which are linear in ΔP and polynomial on dead oil viscosity, bubble point viscosity and bubble point

pressure. More recently, , Dindoruk and Christman (2004) [30] generated a model linear in ΔP with the slope depending exponentially in ΔP as well. Hossain (2005) [31] used a dataset of heavy oils to modify Beal's equation.

The detailed formulas of the correlations discussed are presented in Table 1. Those methods have been thoroughly reviewed and the comparison of their performance against the dataset utilized in this work has been carried out in [5]. It is recommended that the reader goes through this reference for a more comprehensive and quantitative review of the correlation-based methods and to compare against the data-driven models presented in this paper.

Table 1: Correlations to predict μ_{ob}

Beal (1946)	$\mu_o = \mu_{ob} + 10^{-5} \cdot (P - P_b)\left(2.4 \cdot \mu_{ob}^{1.6} + 3.8 \cdot \mu_{ob}^{0.56}\right)$	Kouzel (1965)	$\mu_o = \mu_{ob} \cdot \exp(\alpha(P - P_b))$ $\alpha = 5.50318 \cdot 10^{-5} + 3.77163 \cdot 10^{-5} \mu_{ob}^{0.278}$
Vazquez and Beggs (1976)	$\mu_o = \mu_{ob}\left(\frac{P}{P_b}\right)^m$ $m = C_1 \cdot P^{C_2} \cdot \exp(C_3 + C_4 P)$ $C_1 = 2.6$, $C_2 = 1.187$, $C_3 = -11.513$, $C_4 = -8.98 \cdot 10^{-5}$	Labedi (1982 Libya)	$\mu_o = \mu_{ob} + \frac{10^{-2.488} \cdot \mu_{od}^{0.9036} \cdot P_b^{0.6151}}{10^{0.01976 \cdot \gamma_{API}}}(P/P_b - 1)$
Labedi (1982 Nigeria)	$\mu_o = \mu_{ob} + 0.0483 \cdot \mu_{od}^{0.7374}(P/P_b - 1)$	Khan (1987)	$\mu_o = \mu_{ob} \cdot \exp\left(9.6 \cdot 10^{-5}(P - Pb)\right)$
Al-Khafaji (1987)	$\mu_o = \mu_{ob} + 10^F$ $F = -0.3806 - 0.1845 \cdot \gamma_{API} + 0.004034 \cdot \gamma_{API}^2 - 3.716 \cdot 10^{-5} \cdot \gamma_{API}^3 + 1.11 \log_{10}(0.07031(P - P_b))$	Abdul-Majeed (1990)	$\mu_o = \mu_{ob} + 10^{G - 5.2106 + 1.11 \cdot \log_{10}(6.894757(P - P_b))}$ $G = 1.9311 - 0.89941 \ln(R_s) - 0.001194 \cdot \gamma_{API}^2 + 9.2545 \cdot 10^{-3} \cdot \gamma_{API} \cdot \ln(R_s)$
Petrosky (1990)	$\mu_o = \mu_{ob} + 1.3449 \cdot 10^{-3}(P - P_b) \cdot 10^{X_2}$ $X_1 = \log_{10}(\mu_{ob})$ $X_2 = -1.0146 + 1.3322 \cdot X_1 - 0.4876 \cdot X_1^2 - 1.15036 \cdot X_1^3$	Kartoatmodjo and Schmidt (1991)	$\mu_o = 1.00081\mu_{ob} + 1.127 \cdot 10^{-3} \cdot (P - P_b) \cdot \left(-6.517 \cdot 10^{-3} \cdot \mu_{ob}^{1.8148} + 0.038 \cdot \mu_{ob}^{1.59}\right)$
Orbey and Sandler (1993)	$\mu_o = \mu_{ob} \cdot \exp(\alpha(P - P_b))$ • Parriffinic hydrocarbons $\alpha = 6.76 \cdot 10^{-5}$ • Akylbenzes and cyclic hydrocarbons $\alpha = 7.24 \cdot 10^{-5}$ • Average $\alpha = 6.89 \cdot 10^{-5}$	De Ghetto (1994)	Extra heavy oil $\gamma_{API} \leq 10$: $\mu_o = \mu_{ob} + 10^{-2.19} \cdot \mu_{od}^{1.055} \cdot P_b^{0.3132}/10^{0.0099 \cdot \gamma_{API}} \cdot (P/P_b - 1)$ Heavy oil $10 \leq \gamma_{API} \leq 22.3$: $\mu_o = -0.9886\mu_{ob} + 2.763 \cdot 10^{-3} \cdot (P - P_b)\left(-11.53 \cdot 10^{-3} \cdot \mu_{ob}^{1.7933} + 0.0316 \cdot \mu_{ob}^{1.5939}\right)$ Medium oil $22.3 \leq \gamma_{API} \leq 31.1$: $\mu_o = \mu_{ob} + 10^{-3.8055} \cdot \mu_{od}^{1.4131} \cdot P_b^{0.6957}/10^{0.00288 \cdot \gamma_{API}} \cdot (P/P_b - 1)$ Agip: $\mu_o = \mu_{ob} + 10^{-1.9} \cdot \mu_{od}^{0.7423} \cdot P_b^{0.5026}/10^{0.0243 \cdot \gamma_{API}} \cdot (P/P_b - 1)$
Kouzel API modified (1997)	$\alpha = -2.34864 \cdot 10^{-5} + 9.30705 \cdot 10^{-5} \mu_{ob}^{0.181}$	Almehaideb (1997)	$\mu_o = \mu_{ob}\left(\frac{P}{P_b}\right)^m$ $m = 0.134819 + 1.94345 \cdot 10^{-4} \cdot R_s - 1.93106 \cdot 10^{-9} \cdot R_s^2$
Elsharkawy and Alikhan (1999)	$\mu_o = \mu_{ob} + \frac{10^{-2.0771}(P - P_b)\mu_{od}^{1.19279}}{\mu_{ob}^{0.40712} P_b^{0.7941}}$	Dindoruk and Christman (2004)	$\mu_o = \mu_{ob} + \alpha_6(P - P_b)10^A$ $A = \alpha_1 + a_2 \log_{10} \mu_{ob} + \alpha_3 \log_{10} Rs + \alpha_4 \mu_{ob} \log_{10} R_s + \alpha_5(P - P_b)$ $\alpha_1 = 0.776644115$, $\alpha_2 = 0.987658646$, $\alpha_3 = -0.190564677$, $\alpha_4 = 0.009147711$, $\alpha_5 = -0.000019111$, $\alpha_6 = 0.00006334$
Hossain (2005)	$\mu_o = \mu_{ob} + 0.004481(P - P_b) \cdot \left(0.555955\mu_{ob}^{1.068099} - 0.527737\mu_{ob}^{1.063547}\right)$		

3 Dataset Quality Control (Q/C) and variables selection

To initiate the development of the data-driven models, viscosity measurements were collected from a variety of sources, including published literature and in-house measurements. The dataset comprised of approximately 500 fluids with varying characteristics, ranging from very light oils to heavy ones, originating from various locations around the world.

During the dataset quality control procedure, no fluid curves with an inconsistent viscosity shape versus pressure (i.e., non-increasing) were identified, as all data points for each fluid were continuous and smooth. Among the entire dataset, 89% of the fluids had a bubble point viscosity value of less than 10 cp, 7.8% ranged between 10 and 50 cp and only 3.2% had a value greater than 50 cp (up to 1,760 cp). As very high viscosity oils tend to behave entirely differently from regular oils, such datasets were removed to avoid introducing bias into the models.

While running the Q/C process, only a handful of temperature, GOR and API values were found to be missing and subsequently, fluids with incomplete measurements were entirely removed from the dataset. However, the unavailability of dead oil viscosity values (i.e., μ_o @ $P_{\text{atm}}, T_{\text{res}}$) for most of the fluids' PVT reports was severe as this measurement is often skipped in a PVT study. To provide such critical information to the ML models, it was decided to introduce reservoir temperature as a substitute. This is justified by the fact that temperature is a correlating variable in every dead oil viscosity correlation explored, in conjunction to other inputs already considered (such as API gravity), including the most commonly used one by Beggs and Robinson [32]:

$$\mu_{od} = 10^x - 1$$
$$x = 10^{3.0324 - 0.02023 \cdot API} \cdot T^{-1.163}$$

The range spanned by the Q/C'ed dataset's undersaturated fluids properties is given in Table 2 and their distribution is further illustrated in the histograms in Figure 1 where the vertical axis corresponds to the percentage of each values bin.

Table 2: Database fluid properties range

Parameter	**Minimum**	**Maximum**
μ_{ob} (cp)	0.04	48
μ_o (cp)	0.04	100
GOR (scf/stb)	0	17,118
API	14	57
T $(^\circ F)$	87	376
P_b (psi)	36	7,303
P (psi)	36	12,750

Based on the distribution of P_b values in Figure 1, the utilized fluid database contains reservoir fluids with saturation pressures ranging from less than 100 psi to as high as 7,300 psi, reflecting the dataset's fluid types variability. The fluids' volatility, as represented by their GOR, ranges from almost zero for fluids dominated by their heavy end fraction to as high as 17,000 scf/stb for near-critical, highly volatile oils. Similarly, the API gravity values of the fluids range from 14 to 50+. Furthermore, the reservoir temperature, which significantly influences viscosity, varies widely from less than $100^\circ F$ to over $350^\circ F$.

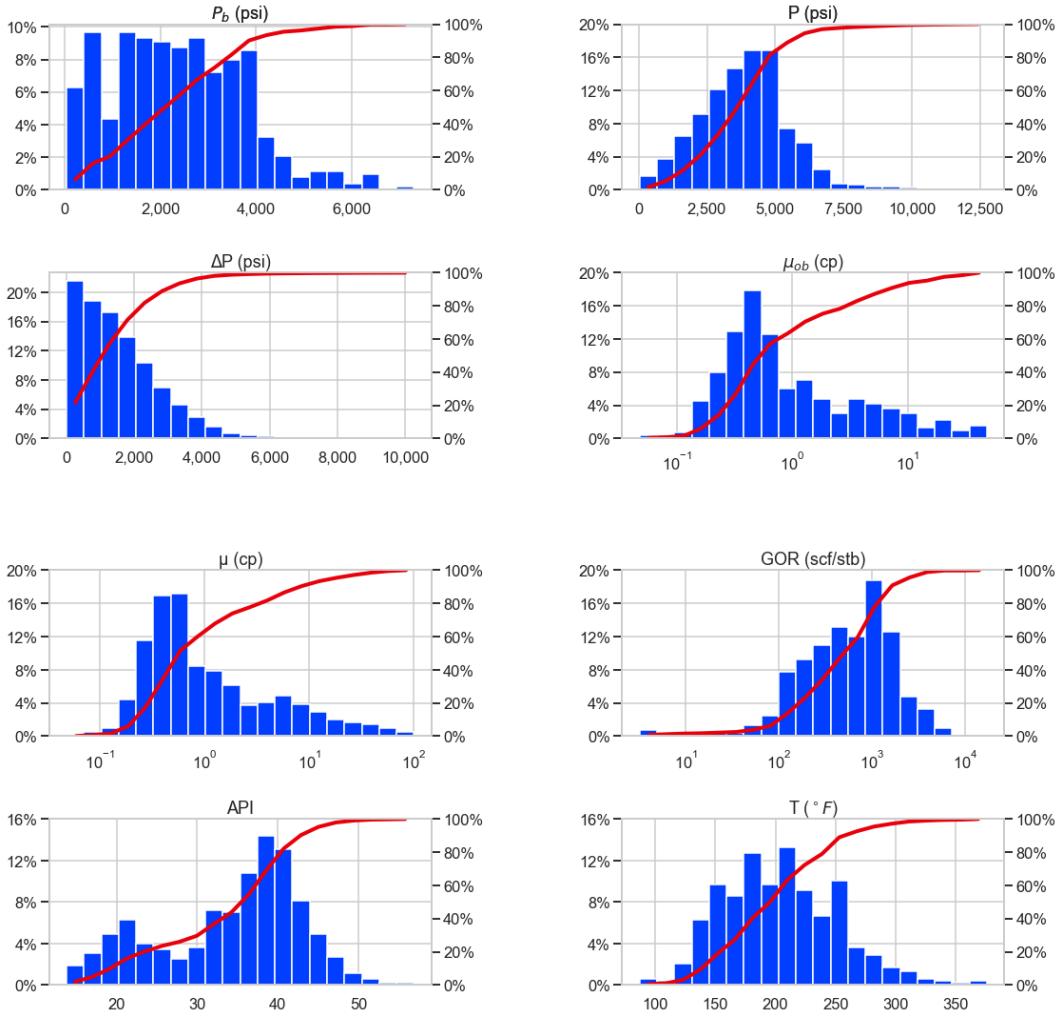

Figure 1: Fluid properties values' probability (in blue) and cumulative histograms (in red)

To facilitate its study, the dataset was further split into four groups based on the bubble-point viscosity value, as depicted in Figure 2. The first and second ranges (0-1 cp and 1-5 cp) can be classified as low to moderately viscous fluids, while the third and fourth ones (5-20 cp and 20-50 cp) correspond to highly viscous fluids that are more difficult to flow. The number of fluids within each range is presented in the histogram at the bottom of the Figure. The box plots above demonstrate the distribution of each input in each range, with the red line signifying the average value and the edges of the box representing the 25th and 75th percentiles. The whiskers extending beyond the box denote the minimum and maximum values of each property. As anticipated, when the bubble-point viscosity increases, API, GOR, P_b and temperature decrease. The interquartile ranges may sometimes be high, but this is a common occurrence when dealing with PVT values in real-world reservoir oil datasets of global origin and various types. The variation in these properties reduces in the high viscosity ranges, although this might be partially attributed to the smaller number of samples in those ranges.

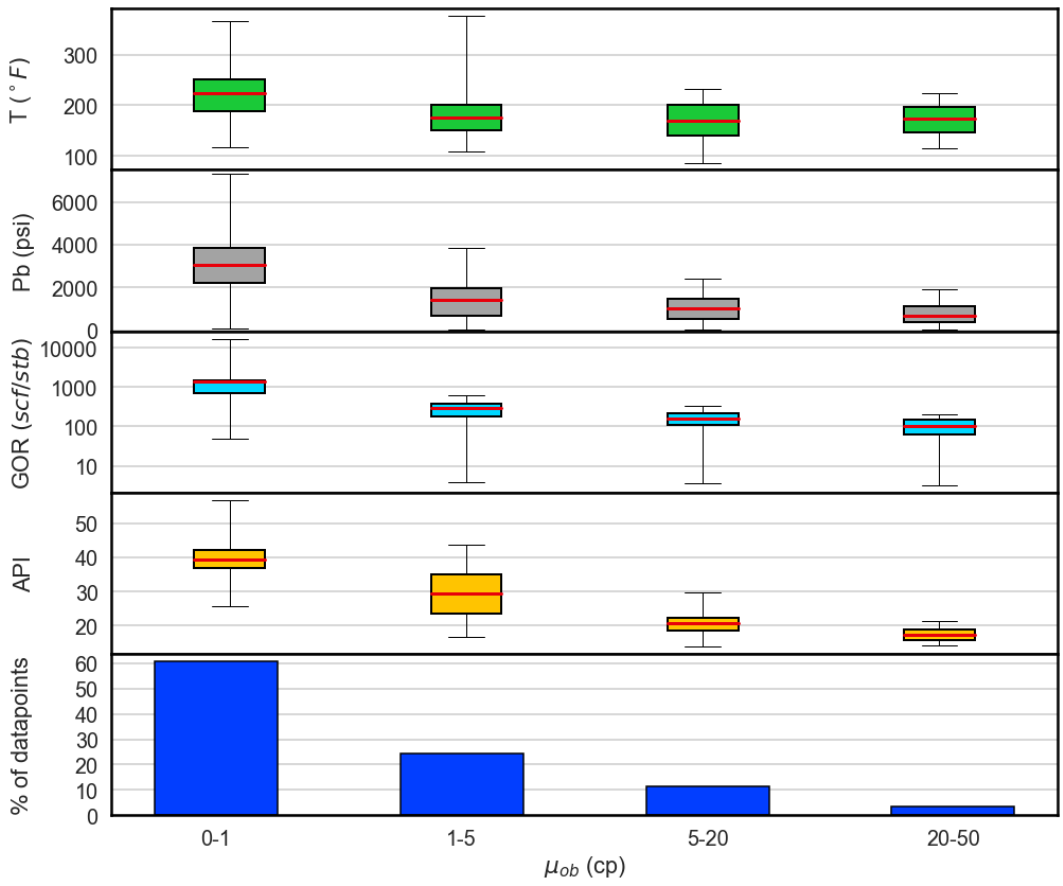

Figure 2: Fluid properties values box plots for four viscosity value ranges

For the ML models input, a fair selection process was used to choose the most appropriate properties out of all available ones. Pressure, GOR, API gravity and temperature were the properties selected as inputs to the models. Bubble point density, heavy end properties, molecular weight and composition were deliberately left out, while dead oil viscosity's contribution was introduced indirectly, as explained before. The reason for excluding these properties is twofold. Firstly, composition and heavy end properties are typically obtained from the compositional analysis as part of a full PVT report for which, it is customary that the field operator has already asked for several undersaturated oil viscosity measurements thus relieving the need for estimations by means of numerical models. Therefore, including detailed information like composition could improve the data-driven models accuracy, but this would provide no added value to the PVT information. This brings up the second reason for the exclusion of those properties, which is comparative fairness. One goal of this research is to demonstrate the ability of ML techniques to automatically capture underlying relations between the input variables and the predicted output, considering the same inputs utilized in the correlation-based models. The latter are usually the result of a researcher's insight when spotting 2 or 3 dimensional patterns in the dataset and applying regression analysis. In this work we investigate whether ML methods are as capable or even better in performing this task using automated, straightforward workflows.

For the ML models output, to avoid introducing bias to the model training process due to the varying number of pressure steps between P_b and P_i per fluid (which is defined arbitrarily by the PVT lab client) the available measurements were interpolated with fluid-specific first order rational functions of pressure. Subsequently, each function was resampled at a fixed

number of pressure values, equally spaced between P_b and P_i, as shown in Figure 3.

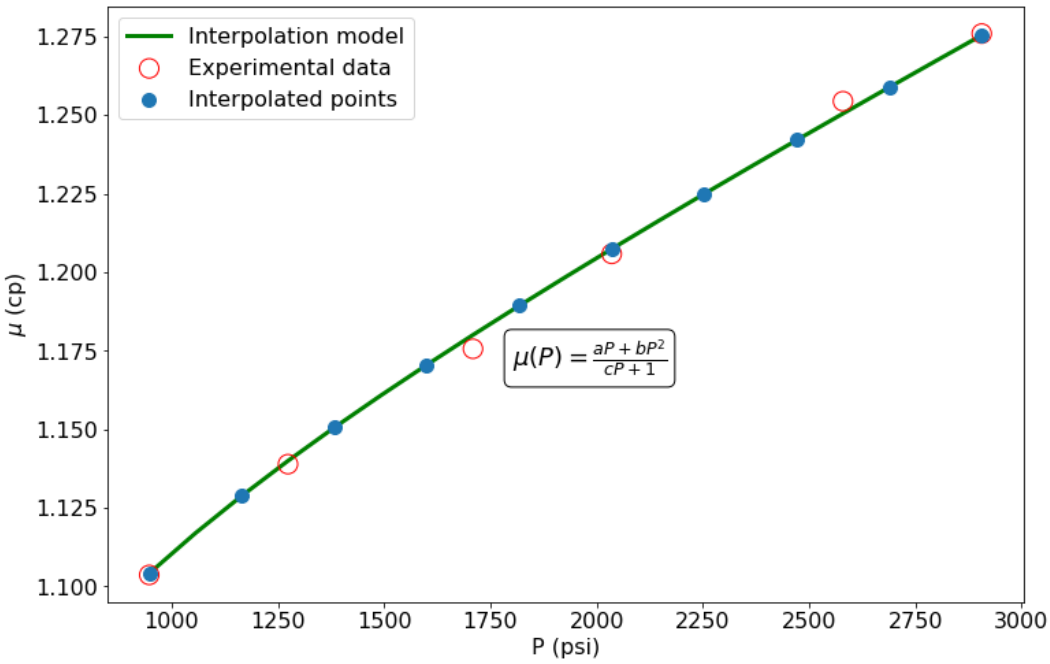

Figure 3: Example of experimental data resampling by means of interpolation (green line)

The Q/C workflow described in this section, entailed tasks that lie within the domain of reservoir engineering expertise. It has involved choosing appropriate input variables, identifying and removing questionable datasets and resampling the output data, all of which align with the methodology employed manually by reservoir engineers. Subsequently, data science-specific techniques will be utilized to further refine the analysis.

4 Development of the ML models

Feature engineering was applied to identify the appropriate combination and form of variables to be introduced in the algorithms' training. Rather than predicting oil viscosity directly, it was decided to train the models to predict the ratio of viscosity to its bubble point value, thereby avoiding problems that arise when the target values span over several orders of magnitude. Various non-linear transformations were carried out to each input to maximize the Pearson's correlation coefficient (discussed in detail below) with the target variable and form the original feature set. Furthermore, the selected features were combined polynomially up to quadratic terms to obtain the augmented feature set, in which the total number of features is $n(n+3)/2$ where n is the number of features. This policy is quite common when building ML models so as to transfer part of the model function complexity from the model itself to the input, thus simplifying the learning process and improving the chances to build accurate predictive models with enhanced generalization capabilities [33]. Regression algorithms were then applied both to the original feature set and the augmented one.

Conventional correlations involving many parameters usually suffer from overfitting as shown in [5] especially when they exhibit high locality, thus downgrading their generalization capability against unseen datasets, such as the ones utilized in this work. On the other hand, ML methods are much more competent in capturing patterns and generating models with

many input parameters, but they also tend to overfit. To avoid overfitting, this study followed a methodical approach of selecting the best models by not only considering their out-of-sample (data it was not trained on) performance, but also requiring small variance between out-of-sample and in-sample (data it was trained on) performance. Furthermore, the diverse sources of the dataset benefit the generalization capabilities of trained models since they don't risk overfitting due to the high locality of the data as seen in correlation-based methods [5]. Notable ML techniques, popular in the ML community were applied and optimized. The selected methods are only briefly presented below and the interested reader may refer to various sources for detailed information on their development, operation and efficiency (for example [34], [35], [36], [37], [38], [39], [40], [41], [42])

4.1 Linear and polynomial regression with regularization

Linear regression aims at fitting a linear in its coefficients w_i and in its features x_i model of the form $\hat{\mu}_o = \sum_i w_i x_i$ to minimize the sum of squared residuals between the observed targets μ_o and those predicted by the linear model $\hat{\mu}_o$. The optimal w_i values minimizing the data fit error, are obtained by a closed-form solution known as the normal equation [34]. Furthermore, various modifications can be used to introduce regularization and to avoid overfitting. Ridge Regression [40] adds a regularization term to the cost function, which corresponds to the sum of the squares of the coefficients $\sum_i w_i^2$. This penalty term forces the model to not only fit the data but also keep the coefficients as small as possible, thus reducing the risk of overfitting. Lasso Regression [41], on the other hand, uses a regularization term that corresponds to the sum of the absolute values of the coefficients $\sum_i |w_i|$ i.e., their first power rather than the second one. This method is particularly useful when dealing with high-dimensional input data, as it can perform feature selection and force specific coefficients to zero, effectively removing them from the model input. The Lasso cost function destroys the closed form solution of the optimal w_i values and the neeed of an iterative optimization method called Stochastic Gradient Descent [42] arises. The Elastic Net technique is a combination of Ridge and Lasso regression that introduces weighted versions of both regularization terms to the error function. It can balance between the two methods and appears to be very effective when multiple features, highly correlated to each other are used.

4.2 Support Vector Regression (SVR)

Unlike traditional regression algorithms which try to minimize the error between the predicted and actual values, Support Vector Regression (SVR) [43] aims to find a hyperplane (i.e., a linear model) which exhibits the maximum possible margin, defined as the distance between the hyperplane and its closest data points, also known as support vectors. This task translates to the minimization of the model weights, therefore it acts as a regularization term. The regression task is accomplished by requiring that the data points beyond the margin lie as close as possible to the hyperplane. However, unlike regular linear modelling, their distance is penalized linearly and only beyond an "ϵ-insensitive tube" which inscribes the hyperplane. This way, SVR ignores points close to the hyperplane while paying (linear) attention only to the ones lying farther. The SVR training maps into a constrained quadratic optimization problem that involves minimizing the error subject to a set of inequality constraints. Non-linear relationships can be handled by transforming the input data into a higher-dimensional space using kernel functions which map the original input data into a new feature space where the data may become linearly separable [44]. Kernel functions such as the linear, the polynomial and the radial basis function (RBF) are the most commonly used [33].

4.3 Ensemble methods

Ensemble learning methods [38] are powerful techniques in ML which leverage the strengths of multiple individual models to achieve better predictive performance. By combining the predictions of several base models, ensemble methods can often achieve superior results compared to a single model alone. The main idea is to exploit the diversity among the base models to reduce overfitting, improve generalization and to increase the model's robustness. Three popular ensemble methods are Random Forest, Gradient Boosting and Adaptive Boosting or AdaBoost. Random Forests (RF) [45] is an ensemble method based on Decision Tree (DT) [46] models. It constructs multiple small size DTs during training and combines their predictions by averaging the output. The trees are typically shallow with just a few layers, which renders them as weak learners. The key to RF's success lies in its ability to create diverse trees by using a random subset of features and bootstrapped samples (sampling with replacement) for each tree. This randomness ensures that the trees are uncorrelated, which helps to reduce the overall variance and improve the generalization capability of the model.

Gradient Boosting (GB) [47] is another ensemble method that builds small size DTs sequentially. Unlike RF, each tree focuses on correcting the residual errors of the previous tree rather than promoting diversity through randomness. At each iteration, a new DT is fitted to the residual errors of the current ensemble and its output is added to the existing prediction. Just as in the case of Random Forests, the trees are shallow and considered weak learners. The final prediction is a weighted sum of the weak learners, where the weights are determined by a learning rate and the improvement in the loss function. GB is particularly powerful in handling complex data and often outperforms other algorithms, but it can be more prone to overfitting if not properly tuned.

Adaptive boosting (AB) [48] is an ensemble method that also trains weak learners sequentially, but with a different approach to weighting and resampling the training data. In each iteration, AB assigns a weight to each training example based on the previous weak learner's performance. Badly predicted examples receive higher weights, which encourages the next weak learner to focus on the harder to learn examples. The weak learners are then combined into a final prediction using a weighted majority vote, where the weights are determined by the weak learner's accuracy. This adaptive process helps AB to focus on the most challenging parts of the data and improve the overall performance.

Voting regression [49] is a ML technique that combines the predictions of multiple regression models to make a final prediction. Each individual model predicts a numeric value and the final prediction is obtained by averaging the predictions of all the models. This technique is used in many popular algorithms such as RFs. Generally it is used in ensemble learning to improve the accuracy and robustness of the prediction. The individual models used in voting regression can be of different types and may use different algorithms or hyperparameters.

In summary, ensemble methods such as RF, GB and AB are powerful techniques that can achieve strong predictive performance by combining multiple models and exploiting their diversity. They have been shown to perform well in a variety of regression tasks, particularly with complex data. However, it is essential to understand the underlying principles and strengths of each method, as well as the importance of proper hyperparameter tuning, to achieve best results.

4.4 Neural Networks

Neural Networks (NN) [37] consist of multiple interconnected layers of nodes, where each node applies a non linear transformation to the input received before passing the result to the next layer, thus bringing in mind the way the human brain operates. NNs are trained using a technique called backpropagation, which adjusts the weights of the connections between layers based on the difference between the predicted output and the actual output. This process is repeated over a large number of epochs, which is the number of times the algorithm goes through the entire training dataset, corresponding to the iterations of a minimization process. The number of layers, nodes and other parameters of the neural network, known as hyperparameters, can be adjusted to optimize its performance on the specific task.

In this work, the popular Scikit-learn [50] and Tensorflow Keras [51] libraries were utilized to implement the algorithms [36]. The objective was to determine the models' optimal parameters along with the hyperparameter values, for each regression algorithm while preventing overfitting. The dataset was split into ten folds of equal size, followed by resampling, for training and testing purposes. It was crucial for practical purposes to ensure that all pressure points belonging to a certain fluid remained in the same fold. The 10-fold cross-validation procedure was used on those folds with the help of a grid search library to select the best hyperparameters for each model [52].

For every set of hyperparameters considered, the regression models were trained 10 times using a different fold as the test set in each iteration. The subsequent section presents the results derived from out-of-sample evaluations. These evaluations assessed the models using all available data, with the reported outcomes originating exclusively from their out-of-sample performance.

5 Results and Discussion

The evaluation and comparison of the generated models' performance is discussed in this Section and results are presented according to the range split defined in Section 3. The typical metrics utilized in the evaluation process are the Average Relative Error (ARE) as a measure of bias and Absolute Average Relative Error (AARE) as a measure of variance, both defined on a percentage basis as follows:

$$\text{ARE } \% = 100 \cdot \overline{e} = \frac{100}{N} \sum_{i=1}^{N} e_i = \frac{100}{N} \sum_{i=1}^{N} \frac{(\hat{\mu_o})_i - (\mu_o)_i}{(\mu_o)_i}$$

$$\text{AARE } \% = 100 \cdot \overline{|e|} = \frac{100}{N} \sum_{i=1}^{N} |e_i| = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{(\hat{\mu_o})_i - (\mu_o)_i}{(\mu_o)_i} \right|$$

where $\hat{\mu}_o$ is the predicted viscosity value and μ_o is the lab measured target value, N is the number of data points considered and e_i is the prediction error on every point. The bar operator denotes the average value. To evaluate how widely ARE and AARE values are distributed around their means, their standard deviation, SDRE and SDARE respectively

defined by

$$\text{SDRE } \% = \sigma_e = \sqrt{\frac{\sum_{i=1}^{N} (e_i - \overline{e})^2}{N-1}}$$

$$\text{SDARE } \% = \sigma_{|e|} = \sqrt{\frac{\sum_{i=1}^{N} \left(|e_i| - \overline{|e|}\right)^2}{N-1}}$$

is provided as well. Lastly, Pearson's correlation coefficient (R^2) is used as a metric of the match between measured and predicted viscosity values, which however considers only error variance, not their bias, defined by

$$R^2_{X,Y} = \frac{\overline{(\mu_o - \overline{\mu_o})(\hat{\mu_o} - \overline{\hat{\mu_o}})}}{\sigma_{\mu_o}\sigma_{\hat{\mu_o}}}$$

where $\sigma_{\mu_o} = \sqrt{\frac{\sum_{i=1}^{N} ((\mu_o)_i - \overline{\mu_o})^2}{N-1}}$, $\sigma_{\hat{\mu_o}} = \sqrt{\frac{\sum_{i=1}^{N} \left((\hat{\mu_o})_i - \overline{\hat{\mu_o}}\right)^2}{N-1}}$

The full set of all five metrics values is given in Appendix A, Tables 5, 6, 7, 8 for the four distinct viscosity values ranges discussed above, following the policy of similar publications [4]. In Appendix B, the measured viscosity values are shown in parity plots vs the model predicted ones in Figures 8, 9, 10.

To enhance readability and to facilitate straightforward model comparison, the results are split in three distinct groups. Group 1 was selected to allow the side by side comparison of the performance of the most popular ML techniques on the original features set. It contains the hyperparameter optimized models of an SVM, four Ensembles and a Neural Network. Group 2 was formed to evaluate the linear in the weights models alongside regularization techniques on the augmented feature set and contains 4 models. Finally, Group 3 shares same models as in Group 1, this time utilizing the augmented feature set. All three Groups along with their models are presented in Table 3. Table 4 illustrates the settings and optimized hyperparameters values selected.

Table 3: Grouping of models

Group	Model	Model type
Group 1 - Non linear models (original feature set)	Model 1.1	SVR
	Model 1.2	Random Forest
	Model 1.3	Gradient Boosting with Decision Trees
	Model 1.4	Neural Network
	Model 1.5	Adaptive Boosting with Decision Trees
	Model 1.6	Voting with Random Forest and SVR
Group 2 - Linear models (augmented feature set)	Model 2.1	SGD with Elastic Net
	Model 2.2	Ridge regularized
	Model 2.3	Lasso regularized
	Model 2.4	Elastic net regularized
Group 3 - Non linear models (augmented feature set)	Model 3.1	SVR
	Model 3.2	Random Forest
	Model 3.3	Gradient Boosting with Decision Trees
	Model 3.4	Neural Network
	Model 3.5	Adaptive Boosting with Decision Trees
	Model 3.6	Voting with Random Forest and SVR

Table 4: Model hyperparameter selection

Model	Hyperparameters selected
Model 1.1	{'C': 0.1, 'epsilon': 0.01, 'kernel': 'rbf'}
Model 1.2	{'criterion': 'friedman mse', 'max depth': 6, 'estimators': 200}
Model 1.3	{'Base estimator': 'Decision Tree', 'learning rate': 0.01, 'max depth': 6, 'estimators': 500, 'subsample': 1}
Model 1.4	{'activation': 'relu', 'batch size': 64, 'epochs': 10, 'layers': 1, 'neurons': 64,'optimizer': 'adam'}
Model 1.5	{Base estimator': 'Decision Tree', 'learning rate': 0.01, 'loss': 'exponential', 'estimators': 100}
Model 1.6	{'Base estimator 1': 'Random Forest', 'Base estimator 2': 'SVR' }
Model 2.1	{'alpha': 0.001, 'learning rate': 'constant', 'loss': 'huber', 'penalty': 'elastic net'}
Model 2.2	{'alpha': 1.0, 'solver': 'cholesky'}
Model 2.3	{'alpha': 1e-05, 'selection': 'cyclic'}
Model 2.4	{'alpha': 0.0001, 'l1 ratio': 0.7, 'selection': 'cyclic'}
Model 3.1	{'C': 1.0, 'epsilon': 0.001, 'kernel': 'rbf'}
Model 3.2	{'criterion': 'squared error', 'max depth': 6, 'estimators': 100}
Model 3.3	{Base estimator': 'Decision Tree', 'learning rate': 0.01, 'max depth': 3, 'estimators': 300, 'subsample': 1}
Model 3.4	{'activation': 'relu', 'batch size': 64, 'epochs': 10, 'layers': 1, 'neurons': 64,'optimizer': 'adam'}
Model 3.5	{Base estimator': 'Decision Tree', 'learning rate': 0.01, 'loss': 'exponential', 'estimators': 100}
Model 3.6	{'estimator 1': 'Random Forest', 'estimator 2': 'SVR' }

The visual representation of the results is given in Figures 4 to 7 where the average relative error (ARE) and the average absolute relative error (AARE) are plotted for each model, as well as for each viscosity value range. This allows the evaluation of how well the ML models perform on different types of oil. To facilitate the comparison, the data series illustrating the performance of each model in the group are slightly shifted to the right to avoid overlapping.

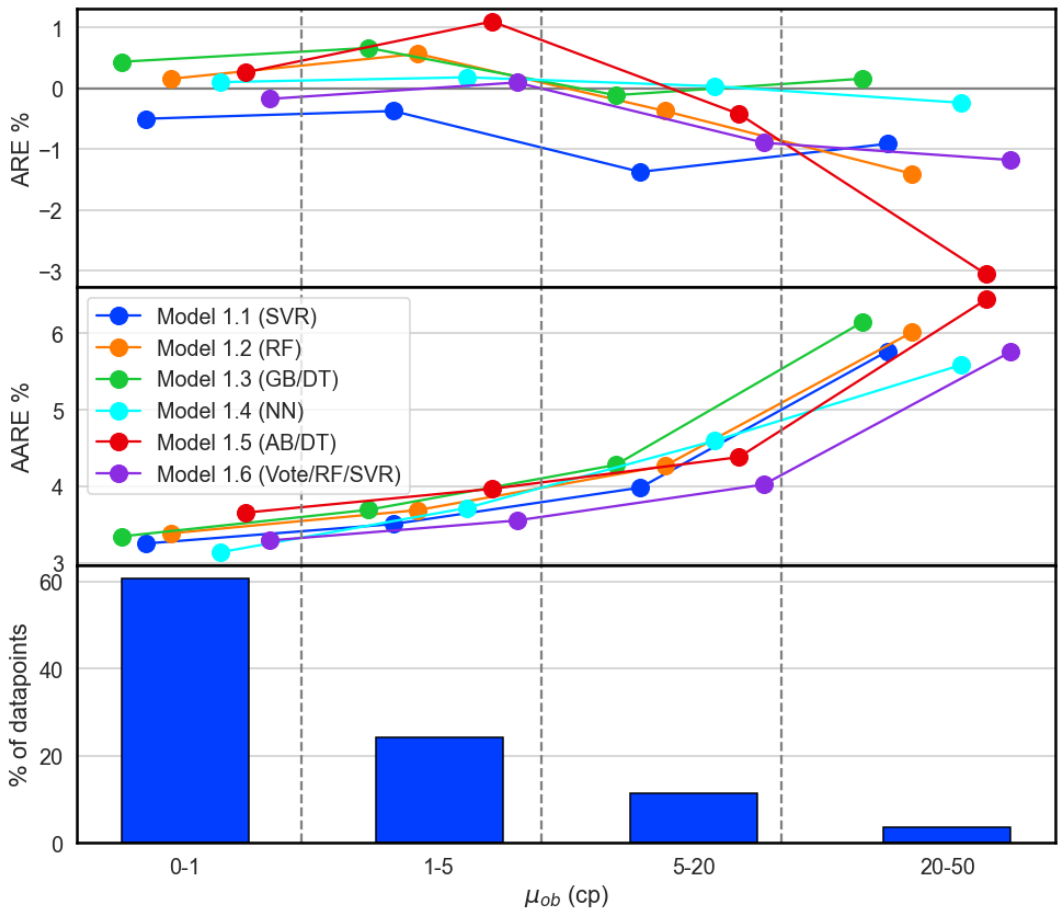

Figure 4: Evaluation of the models in Group 1

Across all four viscosity value ranges in Group 1, Model 1.1 (SVR) systematically underestimates viscosity by 1% on average whereas, Model 1.4 (ANN) appears to be the least biased among the six models in the group, barely surpassing 0.2% at the last viscosity range. Ensemble models exhibit a slight increase in bias when tested against heavier fluids, with the most pronounced case being that of Model 1.5 (AB/DT) which exceeds 3% in the last range, except Model 1.3 (GB/DT), which remains unbiased throughout the tested ranges. As far as the AARE performance is concerned, the reducing number of high viscosity data combined to the increasing span of the high viscosity range explains the poor performance of all six models against very viscous fluids. Side by side comparison of the first (light) and last (heavy) fluid ranges indicates an AARE increase of more than 3%. Similar behavior had been observed when this dataset was used for the evaluation of correlation-based models [5].

The results of the linear in the weights models in Group 2 are presented in Figure 5. Model 2.1 (SGD/EN) demonstrates enhanced capability to avoid bias when tested against volatile oils, however, the other methods outperform it when it comes to heavy oils (around 2% for Models 2.2, 2.3 and 2.4 vs 4% for Model 2.1). Models 2.2 (Ridge) and 2.3 (Lasso) exhibit similar performance, therefore this is also observed with Model 2.4 (Elastic Net), which is a linear combination of the two. When it comes to variance however, AARE values for all four models vary between 4% for the first range and more than 7% in the last one. Clearly, limiting the models to be linear in the weights, drastically reduces their ability to adapt to the irregular highly viscous data points of the dataset, thus ending up with significantly worse statistics in the low volatility range, compared to the models in Group 1.

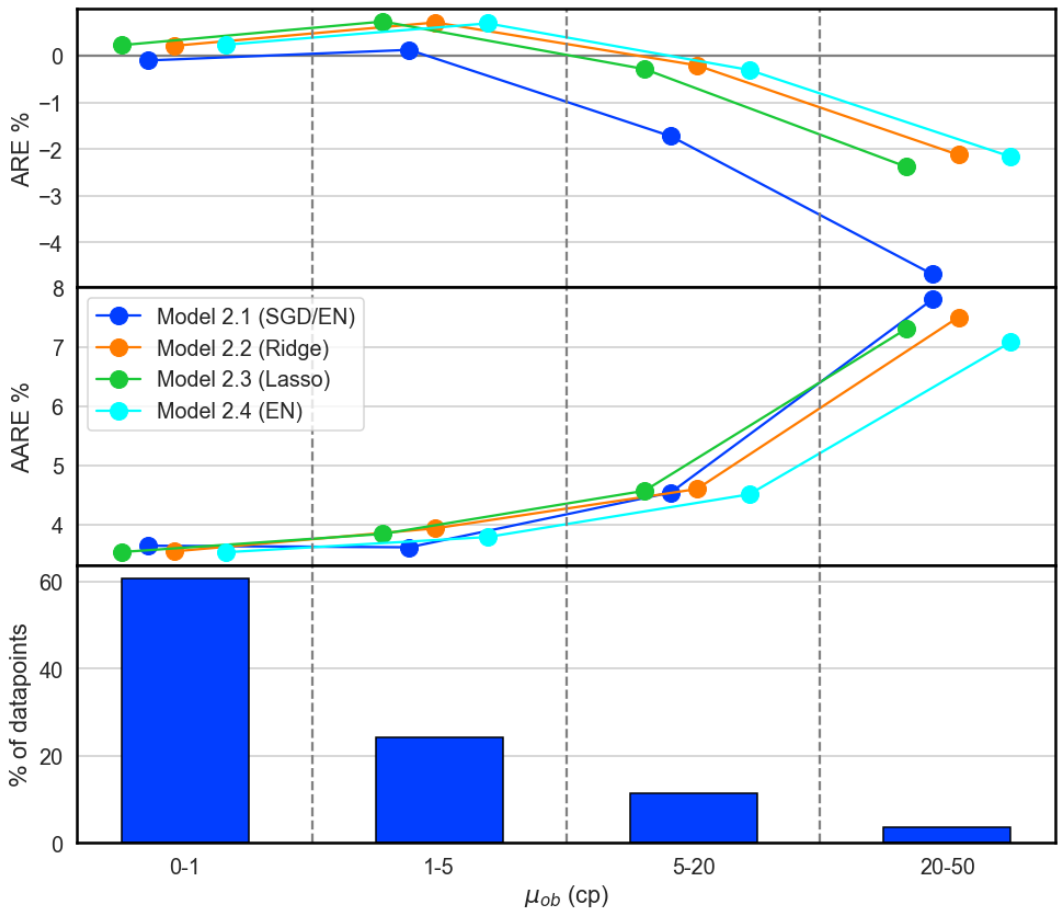

Figure 5: Evaluation of the models in Group 2

For Group 3 (Figure 6), model 3.4 (NN) once again outperforms in terms of bias, this time however it is closely matched by 3.6 (Vote/RF/SVR), both of them not getting over 0.5% for all four ranges. Overall, in terms of variance, the voting model is clearly the best performer of the group in each range, with Model 3.3 (GB/DT) standing out as the less optimal choice with an AARE value close to 5% even at the first range. Models in Group 3, do not outperform those in Group 1 despite the utilization of the augmented feature set on the same models. This must be attributed to the fact that although no new information was introduced to the input by combining the features polynomially, their possibly increased correlation to the output had already been identified by the models training in Group 1. Notably, the majority of the ensemble methods and neural networks used in this study are unbiased, while the support vector machines tend to underestimate their predictions as shown in the performance of models 1.1 and 3.1.

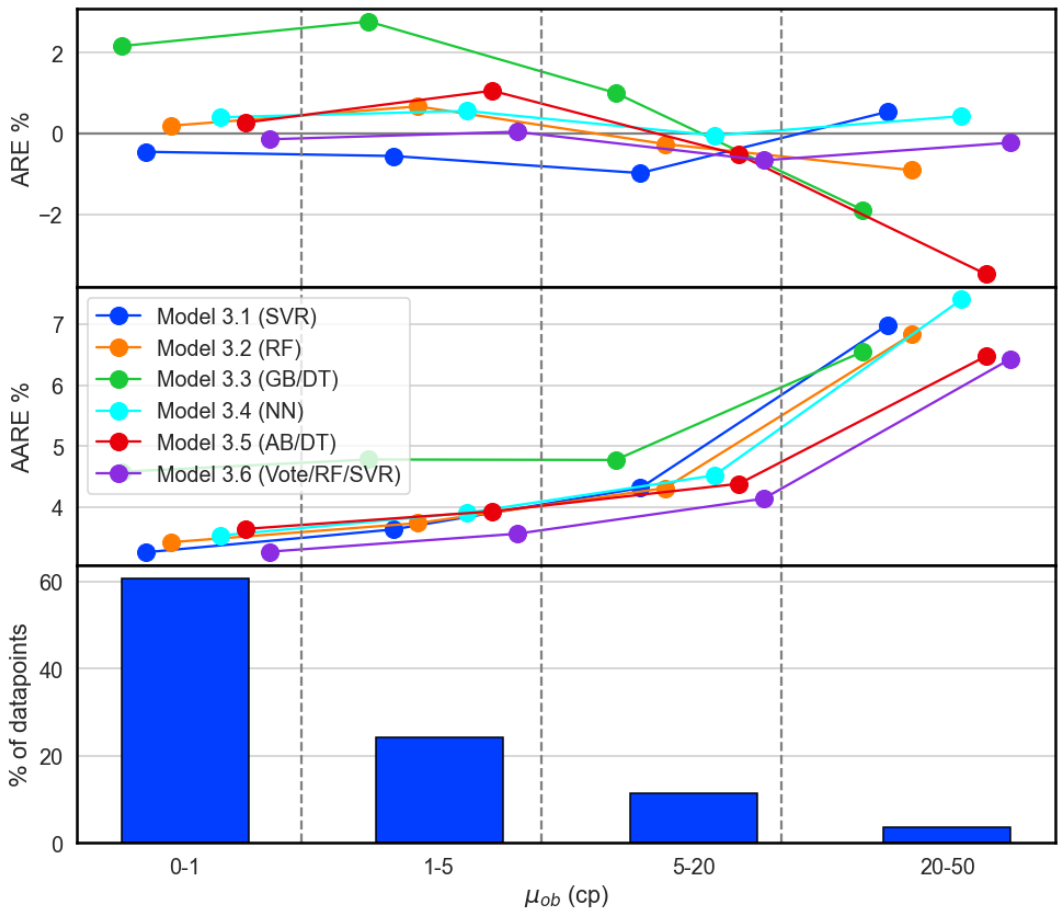

Figure 6: Evaluation of the models in Group 3

Radar charts depicting all correlations AARE have been generated in Figure 7, to enable a more concise comparison of all models. It is worth noting that despite the significant differences in the structure and logic of the tested algorithms, the results are remarkably similar in terms of AARE, except Model 3.3 (GB/DT) which underperforms in the medium volatility oil range as well. The similarity in the results among the tested algorithms could be attributed to the feature transformation process. This process was designed to create features that are highly correlated with the transformed output, which in turn has led to a regularization of the models. Although this regularization helped in achieving optimal results, it may have also limited the ability of the models to capture more complex relationships present in the data.

The scatter (parity) plots constitute a graphical error analysis method. Every point's coordinates in this plot are its experimentally measured value on the horizontal axis and its predicted value on the vertical one. Correlations can be compared on parity plots based on how points are concentrated close to the diagonal. Parity plots to demonstrate the performance of each method are given in Figures 8, 9, 10. The clustering order is followed as in the previous sections. Due to the very wide range of the bubble point viscosity values, the axes are plotted in logarithmic scale. There seem to be some fluids in the μ_{ob} range of (0.01-0.1) that are severely underestimated by many of the models, even the best performing ones. On one hand this can be attributed to the logarithmic scaling of the axes which enhances errors in such values scale, but on the other this range is not within the design area of any of the correlations.

The presented results provide a clear understanding of the benefit of using ML methods to

predict the viscosity of oils, as well as their limitations. The large amount of data available for high and medium volatility oils enables the models to achieve high accuracy in that range. However, for heavy oils, some manually developed correlations in the literature exhibit similar behavior or even perform slightly better than the models generated in this study, as expected when data-driven models with no physics guidance are trained on limited data. In the present case, the learner's training is dominated by the low viscosity data points, thus putting less attention to the highly viscous fluids due to their reduced contribution to the error function. This is further demonstrated by the similarity of the AARE plots in all three groups which share various models and features but same training dataset. Nevertheless, the results also demonstrate that some of the algorithms used, combined with feature transformation techniques, can bridge this performance gap to some extent.

Apart from the methods examined in this work, additional techniques mostly based on the generation of synthetic data to densify the sparsely populated areas, may also be used to improve the lack of uniformity in the dataset [53]. New datapoints can be generated by sampling from the estimated probability density function of the sparse data only. As it is risky, due to the increased dimensionality of the input space, it has to be applied with caution. Alternatively, the Synthetic Minority Oversampling with Gaussian Noise (SMOGN) [54] method can be utilized which provides new datapoints by interpolating between pairs of close points in the poorly populated areas. The very wide viscosity range in those areas renders difficult the task of applying that method without risking the introduction of bias to the enhanced dataset, hence to the developed models. Additionally, synthetic data generation in the undersaturated viscosity context, can be risky, due to the high lab error associated with very viscous fluids. Nevertheless, an engineer could choose any of the methods presented in this study to train a model and it would still perform reasonably well, unlike most of the literature correlations which have limited only applicability.

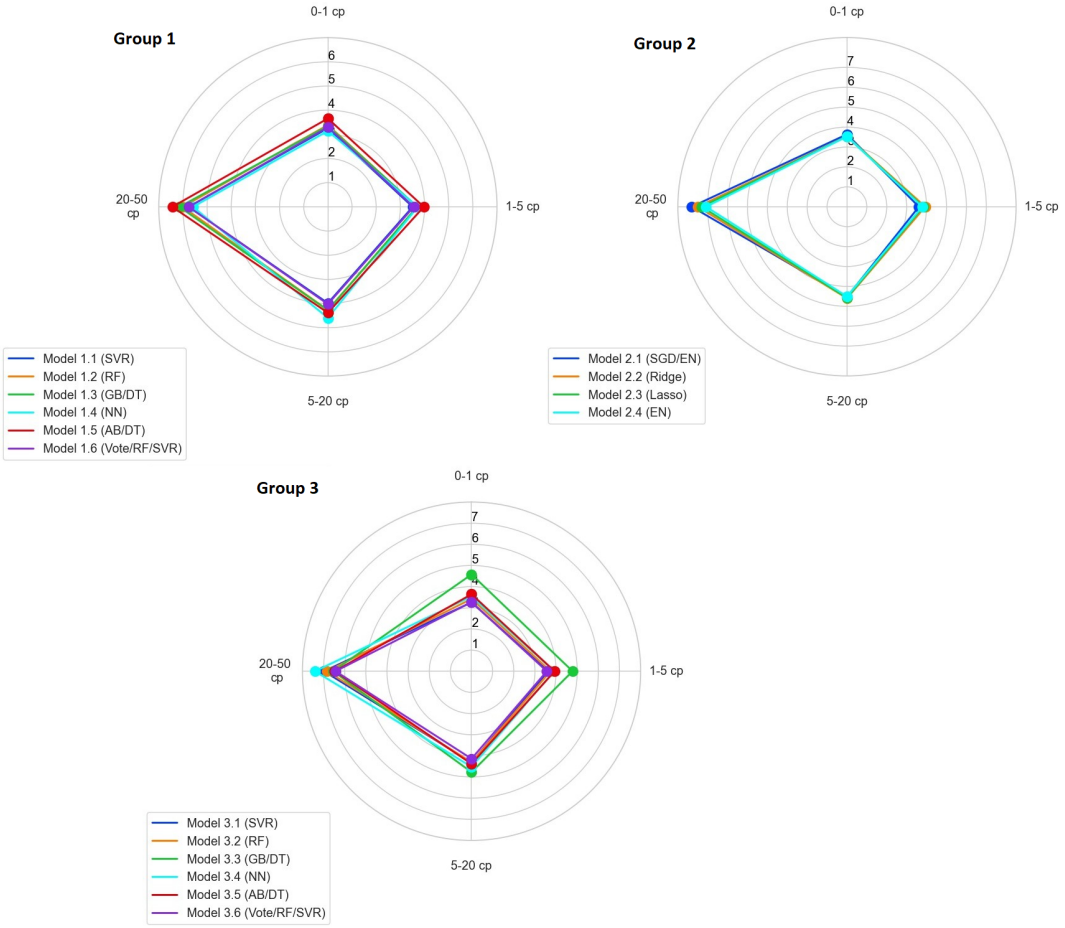

Figure 7: AARE radar charts of all three groups

6 Conclusions

This study presents a comprehensive workflow for a complete ML project aiming at predicting the viscosity of undersaturated oil, followed by a comparison process to ensure that the models were appropriately built and evaluated. Two distinct yet equally important aspects were combined in this study. The first one requires that the expertise of the research team in the field of reservoir and fluid engineering must be recruited, enabling thus critical decisions to be taken. These include the identification of irregular, noisy or problematic data, the selection of related variables from a large set of candidates, the transformation of features and target variables and the appropriate data splitting for training and testing to ensure the integrity of the model training process.

The second aspect involves typical data engineering practices, sometimes called "blind", as they are not taylored to the specific nature of the modeling problem under study. Feature selection and transformation were optimized, the data was scaled and polynomial features were created. Standard data engineering processes were followed and algorithms were carefully selected and tested against a range of hyperparameters to obtain the best possible versions of the selected methods. All in all, the workflow followed is considered a balanced practice regarding both approaches.

Apart from simply building ML models to provide undersaturated oil viscosity trained against

a new dataset, the results of this study demonstrate the effectiveness of combining the two approaches. Deep fluid engineering knowledge needs to be combined to generic-purpose ML models building libraries by incorporating all physical evidence into the developed models. The comparison process used in the study revealed that the tested algorithms produced similar results with only a few exceptions. This suggests that the comprehensive workflow and careful selection of variables, feature transformation and algorithm optimization were successful in producing models that can generalize well to new data. Although we are living in an era where ML is part of everyday life, ML should not be used in a black box fashion. Transparency and interpretability are critical for understanding how ML models make predictions, identifying potential biases or errors and gaining insights into the underlying data and processes.

Acknowledgements: Conceptualization, S.F. and V.G.; methodology, S.F.; software, S.F.; validation, S.F. and V.G.; resources, S.F.; writing—original draft preparation, S.F.; writing—review and editing, S.F., V.G.; visualization, S.F. All authors have read and agreed to the final version of the manuscript.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Abbreviations:

The following abbreviations are used in this manuscript:

AARE	Average Absolute Relative Error
ARE	Average Relative Error
SDRE	Standard Deviation of Relative Error
SDARE	Standard Deviation of Absolute Relative Error
cp	centipoise
psi	pounds per square inch
$^\circ F$	Fahrenheit
API	American Petroleum Institute gravity
scf	standard cubic feet
stb	stock tank barrel

A

Table 5: Detailed metrics values for all models (μ_{ob} = 0-1 cp).

Model	ARE %	SDRE %	AARE %	SDARE %	R^2
Model 1.1	-0.502	5.363	3.255	4.291	0.993
Model 1.2	0.155	5.486	3.388	4.317	0.993
Model 1.3	0.434	5.255	3.347	4.075	0.993
Model 1.4	0.100	5.019	3.140	3.916	0.993
Model 1.5	0.259	5.774	3.657	4.476	0.993
Model 1.6	-0.176	5.401	3.294	4.283	0.993
Model 2.1	-0.105	5.738	3.639	4.437	0.993
Model 2.2	0.208	5.461	3.543	4.160	0.993
Model 2.3	0.222	5.441	3.532	4.145	0.993
Model 2.4	0.233	5.521	3.529	4.252	0.993
Model 3.1	-0.451	5.591	3.258	4.566	0.992
Model 3.2	0.193	5.523	3.422	4.340	0.993
Model 3.3	2.162	5.763	4.578	4.115	0.993
Model 3.4	0.403	5.697	3.533	4.487	0.992
Model 3.5	0.271	5.753	3.640	4.463	0.993
Model 3.6	-0.142	5.447	3.268	4.360	0.993

Table 6: Detailed metrics values for all models (μ_{ob} = 1-5 cp).

Model	ARE %	SDRE %	AARE %	SDARE %	R^2
Model 1.1	-0.375	5.484	3.508	4.230	0.982
Model 1.2	0.564	5.618	3.691	4.272	0.982
Model 1.3	0.662	5.606	3.695	4.267	0.981
Model 1.4	0.178	5.558	3.721	4.132	0.983
Model 1.5	1.096	5.876	3.969	4.468	0.982
Model 1.6	0.094	5.484	3.555	4.176	0.982
Model 2.1	0.122	5.501	3.613	4.150	0.982
Model 2.2	0.709	6.016	3.933	4.607	0.981
Model 2.3	0.729	5.900	3.840	4.537	0.981
Model 2.4	0.689	5.709	3.787	4.327	0.981
Model 3.1	-0.555	5.738	3.632	4.476	0.982
Model 3.2	0.671	5.652	3.736	4.293	0.982
Model 3.3	2.770	5.696	4.779	4.155	0.982
Model 3.4	0.558	5.844	3.905	4.383	0.982
Model 3.5	1.059	5.742	3.928	4.320	0.982
Model 3.6	0.044	5.515	3.561	4.210	0.983

Table 7: Detailed metrics values for all models (μ_{ob} = 5-20 cp).

Model	ARE %	SDRE %	AARE %	SDARE %	R^2
Model 1.1	-1.375	5.894	3.985	4.554	0.988
Model 1.2	-0.375	6.392	4.269	4.771	0.987
Model 1.3	-0.115	6.441	4.280	4.813	0.986
Model 1.4	0.032	6.665	4.595	4.827	0.986
Model 1.5	-0.425	6.280	4.383	4.515	0.987
Model 1.6	-0.897	6.002	4.025	4.540	0.988
Model 2.1	-1.725	6.482	4.540	4.937	0.984
Model 2.2	-0.209	6.855	4.599	5.086	0.986
Model 2.3	-0.290	6.563	4.570	4.718	0.986
Model 2.4	-0.308	6.549	4.509	4.759	0.986
Model 3.1	-0.976	7.004	4.311	5.604	0.985
Model 3.2	-0.259	6.430	4.311	4.776	0.987
Model 3.3	1.005	6.306	4.772	4.241	0.988
Model 3.4	-0.046	6.469	4.520	4.627	0.986
Model 3.5	-0.527	6.263	4.382	4.504	0.987
Model 3.6	-0.661	6.505	4.138	5.062	0.987

Table 8: Detailed metrics values for all models (μ_{ob} = 20-50 cp).

Model	ARE %	SDRE %	AARE %	SDARE %	R^2
Model 1.1	-0.914	8.650	5.764	6.508	0.956
Model 1.2	-1.406	8.243	6.012	5.804	0.961
Model 1.3	0.156	9.154	6.146	6.779	0.955
Model 1.4	-0.239	8.228	5.587	6.040	0.961
Model 1.5	-3.057	8.247	6.440	5.985	0.960
Model 1.6	-1.181	8.186	5.758	5.931	0.960
Model 2.1	-4.678	10.755	7.805	8.749	0.913
Model 2.2	-2.125	10.709	7.507	7.919	0.928
Model 2.3	-2.379	10.156	7.307	7.436	0.936
Model 2.4	-2.164	10.261	7.086	7.723	0.936
Model 3.1	0.537	10.522	6.984	7.881	0.929
Model 3.2	-0.906	9.949	6.832	7.282	0.949
Model 3.3	-1.888	8.812	6.548	6.184	0.957
Model 3.4	0.430	11.631	7.404	8.973	0.929
Model 3.5	-3.478	7.938	6.475	5.754	0.960
Model 3.6	-0.226	9.482	6.432	6.963	0.945

B

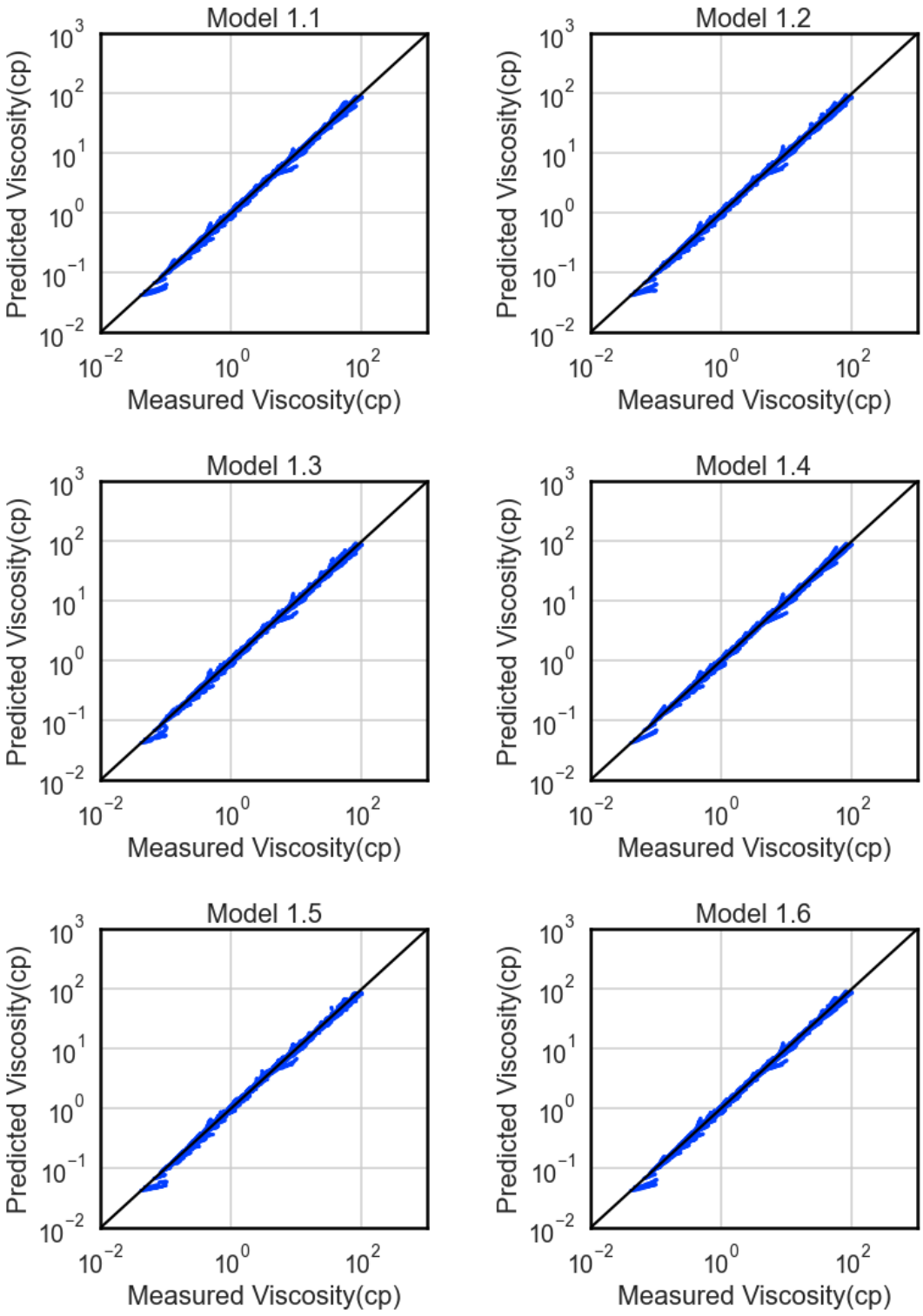

Figure 8: Parity plots of models in Group 1

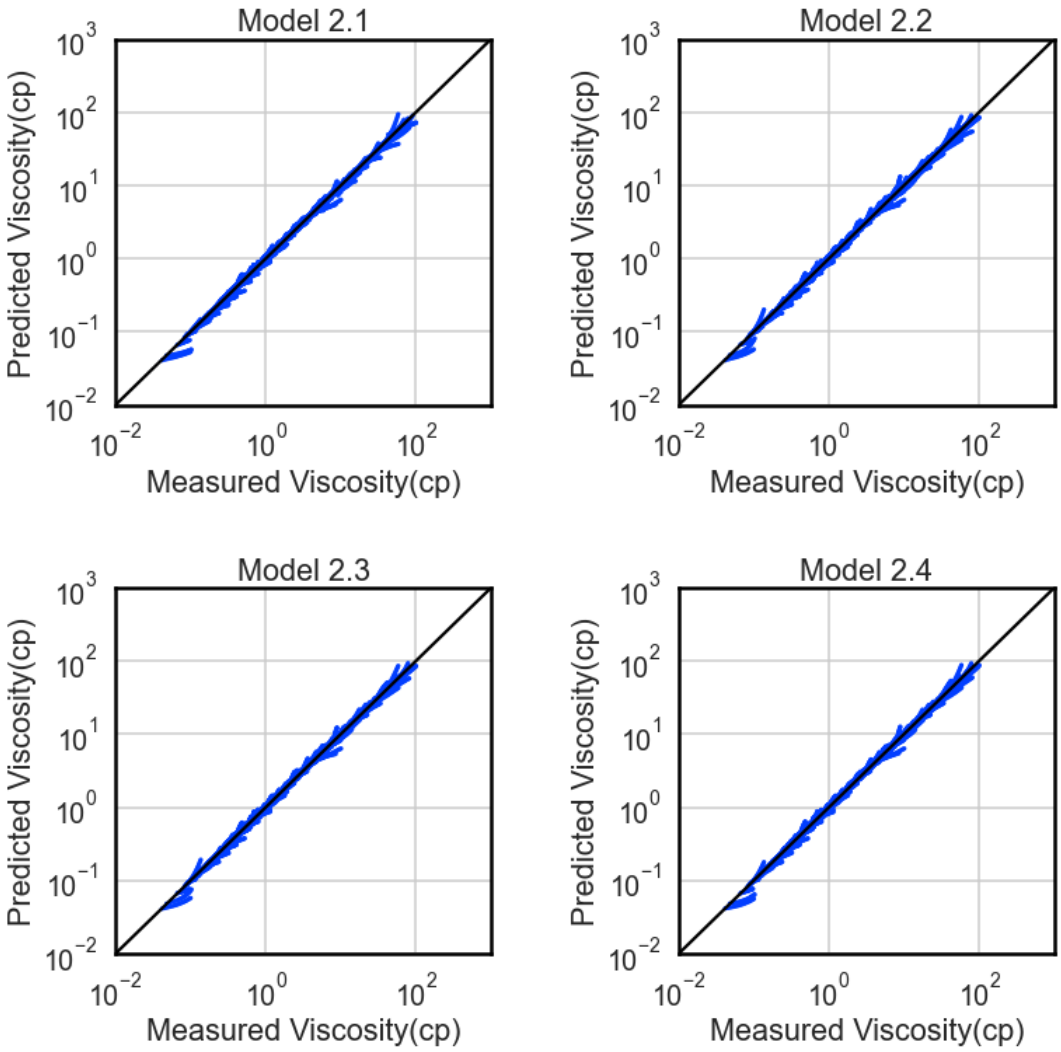

Figure 9: Parity plots of models in Group 2

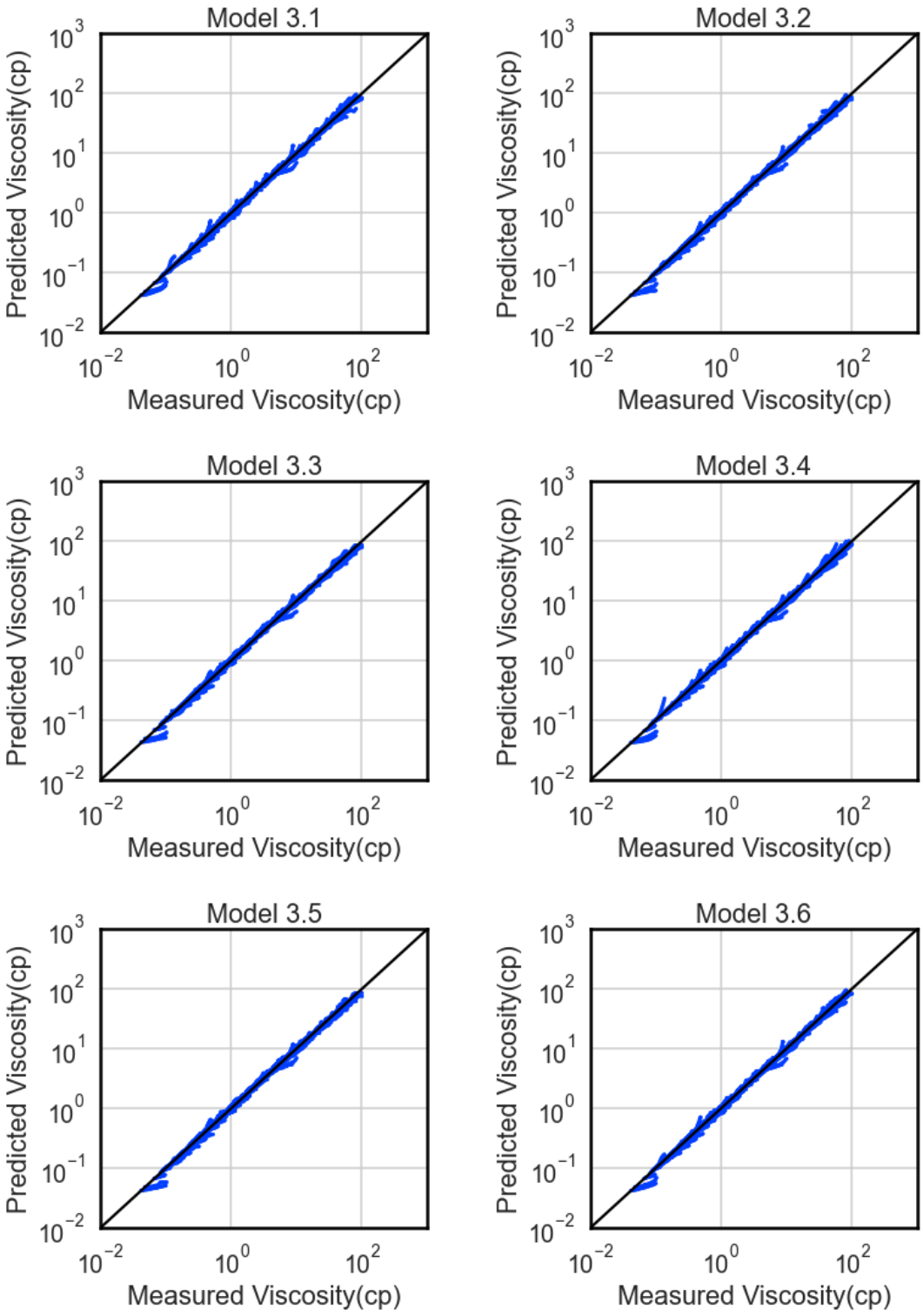

Figure 10: Parity plots of models in Group 3

References

[1] Sawin Kulchanyavivat. *The effective approach for predicting viscosity of saturated and undersaturated reservoir oil.* Texas A&M University, 2005.

[2] Tarek Ahmed. *Equations of state and PVT analysis.* Elsevier, 2013.

[3] Hemanta Mukherjee and James P Brill. Multiphase flow in wells. *Society of Petroleum Engineers of AIME*, 1999.

[4] DF Bergman and Robert P Sutton. Undersaturated oil viscosity correlation for adverse conditions. In *SPE Annual Technical Conference and Exhibition*, 2006.

[5] Sofianos Panagiotis Fotias, Andreas Georgakopoulos, and Vassilis Gaganis. Workflows to optimally select undersaturated oil viscosity correlations for reservoir flow simulations. *Energies*, 15(24):9320, 2022.

[6] Mojtaba Asoodeh and Parisa Bagheripour. Estimation of bubble point pressure from pvt data using a power-law committee with intelligent systems. *Journal of Petroleum Science and Engineering*, 90:1–11, 2012.

[7] Sina Rashidi, Mohammad Mehrad, Hamzeh Ghorbani, David A Wood, Nima Mohamadian, Jamshid Moghadasi, and Shadfar Davoodi. Determination of bubble point pressure & oil formation volume factor of crude oils applying multiple hidden layers extreme learning machine algorithms. *Journal of Petroleum Science and Engineering*, 202:108425, 2021.

[8] Utkarsh Sinha, Birol Dindoruk, and Mohamed Soliman. Machine learning augmented dead oil viscosity model for all oil types. *Journal of Petroleum Science and Engineering*, 195:107603, 2020.

[9] Kassem Ghorayeb, Arwa Ahmed Mawlod, Alaa Maarouf, Qazi Sami, Nour El Droubi, Robert Merrill, Obeida El Jundi, and Hussein Mustapha. Chain-based machine learning for full pvt data prediction. *Journal of Petroleum Science and Engineering*, 208:109658, 2022.

[10] Emad A El-Sebakhy. Forecasting pvt properties of crude oil systems based on support vector machines modeling scheme. *Journal of Petroleum Science and Engineering*, 64(1-4):25–34, 2009.

[11] Daniel Asante Otchere, Tarek Omar Arbi Ganat, Raoof Gholami, and Syahrir Ridha. Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ann and svm models. *Journal of Petroleum Science and Engineering*, 200:108182, 2021.

[12] Ehsan Bahonar, Mohammad Chahardowli, Yaser Ghalenoei, and Mohammad Simjoo. New correlations to predict oil viscosity using data mining techniques. *Journal of Petroleum Science and Engineering*, 208:109736, 2022.

[13] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80:8091–8126, 2021.

[14] Munirudeen A Oloso, Mohamed G Hassan, Mohamed B Bader-El-Den, and James M Buick. Ensemble svm for characterisation of crude oil viscosity. *Journal of Petroleum Exploration and Production Technology*, 8:531–546, 2018.

[15] Petroleum Experts. Ipm suite.

[16] Carlton Beal. The viscosity of air, water, natural gas, crude oil and its associated gases at oil field temperatures and pressures. *Transactions of the AIME*, 165(01):94–115, 1946.

[17] MB Standing and DL Katz. Volumetric and phase behavior of oil hydrocarbon system. *Society of Petroleum Engineers of AIME, Dallas*, 1981.

[18] B Kouzel. How pressure affects liquid viscosity. *Hydrocarbon Processing. March*, 1965:120, 1965.

[19] M Vazquez and HD Beggs. Correlations for fluid physical property prediction. jpt 32 (6): 968–970. Technical report, SPE-6719-PA. DOI: 10.2118/6719-PA, 1980.

[20] Rafa Mohamed Labedi. Pvt correlations of the african crudes. *1980-1989-Mines Theses & Dissertations*, 1982.

[21] SA Khan, MA Al-Marhoun, SO Duffuaa, and SA Abu-Khamsin. Viscosity correlations for saudi arabian crude oils. In *Middle East Oil Show*. OnePetro, 1987.

[22] George E Petrosky. *PVT correlations for gulf of mexico crude oils*. PhD thesis, University of Southwestern Louisiana, 1990.

[23] Trijana Kartoatmodjo, Zelimie Schmidt, et al. New correlations for crude oil physical properties. *paper SPE*, 23556, 1991.

[24] Ali H Al-Khafaji, Ghassan H Abdul-Majeed, Saadia F Hassoon, et al. Viscosity correlation for dead, live and undersaturated crude oils. *J. Pet. Res*, 6(2):1–16, 1987.

[25] Ghassan H Abdul-Majeed, Riadh R Kattan, and Naeema H Salman. New correlation for estimating the viscosity of undersaturated crude oils. *Journal of Canadian Petroleum Technology*, 29(03), 1990.

[26] Hasan Orbey and Stanley I Sandler. The prediction of the viscosity of liquid hydrocarbons and their mixtures as a function of temperature and pressure. *The Canadian Journal of Chemical Engineering*, 71(3):437–446, 1993.

[27] Giambattista De Ghetto and Marco Villa. Reliability analysis on pvt correlations. In *European Petroleum Conference*. OnePetro, 1994.

[28] RA Almehaideb. Improved pvt correlations for uae crude oils. In *Middle east oil show and conference*. OnePetro, 1997.

[29] AM Elsharkawy and AA Alikhan. Models for predicting the viscosity of middle east crude oils. *Fuel*, 78(8):891–903, 1999.

[30] Birol Dindoruk and Peter G Christman. Pvt properties and viscosity correlations for gulf of mexico oils. *SPE Reservoir Evaluation & Engineering*, 7(06):427–437, 2004.

[31] Mohammad Sohrab Hossain, C Sarica, H-Q Zhang, L Rhyne, and KL Greenhill. Assessment and development of heavy oil viscosity correlations. In *SPE International Thermal Operations and Heavy Oil Symposium*. OnePetro, 2005.

[32] H Dale Beggs and John R Robinson. Estimating the viscosity of crude oil systems. *Journal of Petroleum technology*, 27(09):1140–1141, 1975.

[33] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[34] *Normal Equations*, pages 380–382. Springer New York, New York, NY, 2008.

[35] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[36] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019.

[37] James A Anderson. *An introduction to neural networks*. MIT press, 1995.

[38] Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2(1):110–125, 2002.

[39] Raul Rojas. The backpropagation algorithm. In *Neural networks*, pages 149–182. Springer, 1996.

[40] Gary C McDonald. Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1):93–100, 2009.

[41] Jonas Ranstam and JA Cook. Lasso regression. *Journal of British Surgery*, 105(10):1348–1348, 2018.

[42] Nikhil Ketkar and Nikhil Ketkar. Stochastic gradient descent. *Deep learning with Python: A hands-on introduction*, pages 113–132, 2017.

[43] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[44] Tristan Fletcher. Support vector machines explained. *Tutorial paper*, pages 1–19, 2009.

[45] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[46] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[47] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[48] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[49] Joao Mendes-Moreira, Carlos Soares, Alípio Mário Jorge, and Jorge Freire De Sousa. Ensemble approaches for regression: A survey. *Acm computing surveys (csur)*, 45(1):1–40, 2012.

[50] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[51] Jason Brownlee. *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. Machine Learning Mastery, 2016.

[52] Richard R Picard and R Dennis Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.

[53] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[54] Paula Branco, Luís Torgo, and Rita P Ribeiro. Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pages 36–50. PMLR, 2017.