Application of Machine Learning Ideas to Reservoir Fluid Properties Estimation

Chukwuma Onwuchekwa, Shell Petroleum Development Company of Nigeria Limited (SPDC)

This paper was prepared for presentation at the Nigeria Annual International Conference and Exhibition held in Lagos, Nigeria, 6–8 August 2018.

Abstract

Machine learning refers to a range of data-driven techniques that give computers the ability to learn from exposure to data and to make predictions based on the learning. Popular applications of machine learning include hand-written digit recognition technology used by some banks to automatically process cheques, spam filtering technologies used by email applications to detect spam mails and object recognition technologies in self-driving cars, to name a few. Examples from the Oil and Gas sector, though less exotic, have also been growing steadily. For example, artificial neural networks have been used for years for the estimation of reservoir properties such as permeability and porosity; there have also been applications of the technique in the analysis of the huge amount of pressure and flow rate data from permanent downhole gauges; also, data-driven predictive analytics have been applied in mature fields with huge amounts of data.

This paper discusses the results of an investigation of the performance of some machine learning techniques in the prediction of reservoir fluid properties. The techniques investigated include K Nearest Neighbors (KNN), Support Vector Regression, Kernel Ridge Regression, Random Forest, Adaptive Boosting (Adaboost) and Collaborative Filtering. PVT data from a database of 296 oil and 72 gas reservoirs from the Niger Delta were used in the study. The input data used in the training include initial reservoir pressure, saturation pressure, solution gas oil ratio (for oil samples), formation volume factor, condensate gas ratio (for gas samples), API gravity, gas gravity, saturated oil viscosity and dead oil viscosity. Trained models were developed using the techniques and used to predict saturation pressure and formation volume factor, oil viscosity and condensate gas ratio respectively for samples that were not part of the training.

It was found that all six techniques gave very good results for the oil formation volume factor, comparable to and in some cases exceeding the performance of standard industry correlations such as Standing and Vasquez-Beggs. The techniques also gave good results for bubble pressure better than the standard correlations. For oil viscosity, the Random Forest and Adaptive Boosting gave very good results, of the same quality as that obtained with the popular Beggs-Robinson correlation, and did not require dead oil viscosity data. Performance of the techniques in estimating gas PVT parameters was not as good; due perhaps to the limited gas data. However, Adaptive Boosting and Support Vector Regression gave good results for dew point pressures. Overall the results indicate that these machine learning techniques offer promise for fluid properties estimation and should be given consideration where a company has acquired large amount of PVT data in a geological basin it operates.

Introduction

In recent years, there has been accelerated application of Machine Learning and Artificial Intelligence techniques. This upsurge in application have been driven by the huge amount of data available to technology giants such as Google, Microsoft, Facebook, Amazon and others. Some applications of machine learning include page rank algorithms used in search engines such as Google and Bing, recommender systems used by sites such as Amazon and Netflix to recommend products to customers based on earlier purchases or purchases by other customers with similar profile, and applications in computational biology where machine learning has helped in the interpretation of large genomic datasets. The petroleum industry which was a pioneer in the development and application of techniques of learning from data seems to be lagging in this modern upsurge in the application of Machine Learning. Decline Curve analysis and the various correlations we now routinely use are early examples of learning and making predictions from data. There has been a sizeable number of papers and case studies published on the application of machine learning and related techniques in the petroleum industry. However there has yet not been many widely-adopted practices in the same manner that decline curve analysis and the empirical/semi-empirical correlations were adopted.The following is a sample of applications of machine learning in the petroleum industry that has been published in the literature. Early applications focused on the estimation of reservoir properties using artificial neural networks[1,2,3]. Application to other areas has since evolved. Tian and Horne[4] applied machine learning techniques to interpret flow rate, pressure and temperature data from permanent downhole gauges(PDGs). They found that Kernel Ridge Regression, a machine learning technique, can be used to recover full reservoir behaviors (i.e. well bore storage, skin effect, infinite acting radial flow, boundary effects) from the PDG data. Prof. Mohaghegh and his research team[5,6,7] has pioneered and have been pushing the idea of Surrogate Reservoir Models(SRMs). SRM design involve a few intelligent techniques, including machine learning, which result in a surrogate model that mimics the static and dynamic characteristics of a Full Field Model (FFM), but runs in seconds compared to hours for the FFM. Also based on the same ideas, is Top-Down Intelligent Reservoir Modelling(TDIRM)[8] which uses the” big-data” approach to infer the physics of mature reservoirs instead of traditional reservoir simulation. Unlike techniques such as decline curve analysis which rely only on production/injection data, TDIRM uses all available field measurements (production/injection data, well location and trajectories, well logs, bean settings, well activities etc.) to deduce the physical phenomena going on in the reservoir as against the pre-determined fluid flow equations on which traditional reservoir simulators are based. This makes the TDIRM well-suited for modelling of unconventional reservoirs where the physics governing fluid flow are not yet well-understood. TDIRM will also be suitable for management of mature reservoirs near their end-life where a huge finite-difference reservoir modelling and history-matching effort will not be justified. For fluid properties, Moussa et. al.[9] recently developed a hybrid model consisting of artificial neural networks and self-adaptive differential evolution techniques that can predict bubble point pressure and initial solution gas oil ratio using only oil gravity, gas gravity and reservoir temperature.

Fluid properties estimation is an area where machine learning techniques could be useful, and it is the goal of this paper to evaluate the suitability of some machine learning algorithms in estimating reservoir fluid properties. Except for collaborative filtering technique which was chosen because of its ability to handle cases with data gaps in the training data, all the other techniques considered in this paper are simple techniques which do not require any specialized expertise in training the data and for which there are libraries available in programing platforms such as python, R or MATLAB. The data used for the study were taken from a database of PVT data acquired over the years in the Niger Delta by Shell Petroleum Development Company of Nigeria. The dataset had data from 296 oil reservoirs and 72 gas reservoirs. Oil data extracted from the database include depth, initial reservoir pressure, reservoir temperature, bubble point pressure, initial solution gas oil ratio, oil formation volume factor, dead oil viscosity, oil viscosity at bubble point, oil gravity and gas gravity. For gas reservoirs data extracted include depth, initial reservoir pressure, reservoir temperature, dew point pressure, condensate gas ratio, Z factor, gas expansion factor and gas gravity. Of the 296 oil data points, 50 were reserved for testing and not involved in the training of the algorithms. Similarly, 15 of the gas data points were reserved for testing. Apart from the collaborative filtering technique for which a custom python program from the literature was used, the “off-the-shelf” python-based machine learning library -Scikit Learn – was applied for the other algorithms.

The Algorithms

The following are brief descriptions of the machine learning algorithms considered:

K Nearest Neighbor(KNN):

The KNN algorithm is the simplest technique considered. It is like what is done intuitively during the identification of analogue reservoir(s) to a target reservoir based on the closeness of their parameters. KNN algorithm formalizes this approach using distance metrics. There are many distance metrics that can be used but the most common is the standard euclidean distance, which is what is used in this work. The idea is to find K (where K is a user-defined number) data points from the training dataset closest in distance to a new query point. The mean of the parameter of interest (e.g. bubble point pressure) for these K nearest neighbors is then predicted as the value of the parameter for the new point. Note that each data point is an n dimensional vector consisting of the n covariates (e.g. depth, reservoir temperature, initial reservoir pressure, oil gravity, gas gravity etc.,).

Random Forest:

Random Forest algorithm belong to the so-called ensemble methods that combine several "mediocre" base estimators to give better combined performance. In the case of random forest, the base estimator is a simple decision tree estimator. The method is called random forest because random subsets of the training data (with resampling) are trained on random subsets of the features. The result is a set of decision trees based on these subsets of data and covariates. Prediction is obtained from the model by taking the mean of the prediction of the set of trees for the regression case (which is the prediction case of interest in this paper).

Adaptive Boosting (Adaboost):

Adaboost algorithm is another ensemble method. However, unlike the case of random forest where a predetermined number of smaller samples are taken from the training dataset and used in parallel for estimation, adaboost works by sequentially generating estimators from modified versions of the training dataset using a technique referred to as boosting. In boosting, weights are applied to each data point in the training data set, starting with equal weights. In subsequent iterations, those data points that are not well predicted by the estimator generated in a previous iteration are given higher weight and greater focus in the subsequent training. In this way a set of estimators are generated that are experts at predicting certain combinations of the features. For a query data point, prediction is achieved by weighted vote by all the estimators.

Support Vector Regression:

Support Vector Regression is based on the well-known classification algorithm Support Vector Machine. This classification algorithm became popular in the 90's because of its ability to give surprisingly good results that were of same quality as those obtained using artificial neural networks without requiring the extensive (and sometimes mysterious) tuning that neural networks required. It belongs to the class of the so-called Maximum Margin Classifiers and works by maximally separating hyperplanes. It uses the so-called kernel trick which allows for the mapping of original data to higher dimensional space without explicitly defining the higher dimension. It's regression variant – Support Vector Regression – inherits this property and can generate high performing non-linear regressors that take advantage of the kernel-trick.

Kernel Ridge Regression:

Kernel Ridge Regression is another regression algorithm that utilizes the kernel-trick, but this time applied to ridge regression. Ridge regression is simply the application of regularization parameter to linear regression to avoid over-fitting. As Kernel Ridge Regression uses the kernel-trick, it can therefore generate linear or non-linear regression functions in higher dimensional space depending on the kernel function used. Kernel Ridge Regression models are quicker to train compared to Support Vector Regression; but are slower at prediction time.

Collaborative Filtering:

Collaborative filtering is a popular technique used in building recommender systems. It takes advantage of a user's like (or dislike) for products and other users likes (or dislikes) for the same and other products to build a profile of a user's taste to recommend new products to her/him. Consider the table below of movie ratings by different users on a scale of 1 to 5:

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5
User 1		?	4		
User 2	5	?	1	2	4
User 3		3	5		
User 4	4	4	?		5

The question recommender systems seek to answer is: How would a user have rated those movies that he did not rate given his ratings of other movies and given other users ratings of the target and other movies. For example, how will User 4 rate Movie 3? Note that Users 4 and 2 liked Movies 1 and 5; may be User 4 will rate Movie 3 low, like User 2. However, note that User 4 rated all the movies he watched high, maybe he likes movies, also note of all the people that rated Movie 3 only User 2 rated the movie low. In collaborative filtering, these insights will all be considered when probabilistically arriving at the most likely rating of Movie 3 by User 4. Petroleum Engineers are faced with these same situations when we must work with insufficient data. Consider the following data from a fluid properties database:

	Depth ft_ss	Press psia	PB psia	RSI scf/b	Boi v/v	TR deg/f	Oil 60/60f	Gas air=1
Reservoir 1	8462	3674	?	896	1.375	161	?	0.67
Reservoir 2	7665	3449	3349	893	?	160	0.91	0.65
Reservoir 3	7418	3240	?	386	1.136	139	0.94	?
Reservoir 4	10101	4372	4350	1255	1.591	184	0.84	0.70
Reservoir 5	10392	4509	3893	?	1.480	194	?	0.71

We will like to estimate the missing values giving the data that we do have for the reservoir of interest and other reservoirs. In collaborative filtering, Reservoir 1, for example, will “collaborate” with the other reservoirs by contributing its own information. Available information from all the reservoirs are used in a collaborative manner to arrive at estimates for the missing values. In a real database, there will be hundreds or thousands of data points and some of the data may appear contradictory. Some of the contradictory trends may indeed be noise or error in the data; but, they may also be due to latent variables that are not known or cannot be measured such as charge history or biodegradation at some level. Collaborative filtering, can pick up these latent effects (that are not usually obvious to the human eye) from data if sufficient data is collected to establish some pattern.

One algorithm for solving Collaborative filtering problems is Probabilistic Matrix Factorisation(PMF).This is the algorithm used in this paper. For mathematical details of this technique refer to Salahkhutdinov and Mnih[10]. The basic idea is as follows: If we have a matrix R of dimension N x P, where N can be the no of oil reservoirs in our database and P can be measured oil PVT parameters per reservoir. Note that not all P parameters are available for all the reservoirs (that is there are gaps in the data). The goal of the machine learning is to find two matrices U and Q of dimensions N x k and k x P respectively such that we can approximate our matrix R. The dimension k can be viewed as the latent features we want to unravel. Each row of U represents how each reservoir is associated with these latent features and each column represents how our measured fluid parameters are associated with these features. Note that if these two matrices can be successfully learned, then by their multiplication the missing values in R will be filled up.

$$R \approx U \times Q$$

Successful learning consists of iteratively finding these two matrices such that in the resulting R, the original data in the database are “history-matched” within a tolerance, while the gaps in the data are filled.

Application of the Algorithms to Oil PVT

Oil PVT tests were handled in two parts namely, a. Fitting and predicting bubble point pressure and oil formation value factor and b. Fitting and predicting of oil viscosity data. The covariates were depth, initial reservoir pressure and temperature, initial solution gas/oil ratio, oil gravity and gas gravity. For all the methods it was necessary to scale the data as a pre-processing step to bring them to approximately the same order of magnitude. For the PMF technique the following scaling technique was found to be most effective:

Scaled Parameter = (Parameter – Mean)/Standard Deviation

For the other methods, normalization by an appropriate value to bring them to predominantly single digit order of magnitude was found adequate.

PMF Training:

For Collaborative Filtering using the PMF technique, the training matrix R will contain all the covariates listed above in addition to the desired parameters – (i.e. bubble point pressure and oil formation volume factor or viscosity). The probabilistic matrix factorization is performed on the training dataset. One advantage of the PMF technique is that it can tolerate missing data in the training dataset; no reservoir in the training set is omitted for gaps its data. After the training, adjustments are applied to obtain as good a fit of the training data as possible. The test data is then included with the bubble point pressure and oil formation volume factor (or viscosity) for the test data excluded. By the iterative PMF algorithm the omitted test data are then filled by the factorization. This then constitutes the prediction. The adjustment factors obtained during the training are then applied to get the final prediction which is then compared to the actual data.

The python code of Salakhudinov and Mnih[10] for PMF was modified and used in this work. They used a stochastic gradient descent algorithm for optimization. An expectation-maximization (EM) optimization code developed in the course of this work which performed very well for some benchmark datasets from other subject domains(e.g movie rating) did not perform as well as the Salakhudinov and Mnih code for the dataset of this work. It is not clear if this reflects the ability of these algorithms in solving PMF problems or a reflection of the implementations.

Training of the Other Algorithms:

For the other algorithms, training was straightforward. It consists as follows: Fit the training data given the training covariates and predict the test data given the test covariates. These methods require that the selected features for each reservoir be complete or that reservoir be taken out of the training; also, prediction cannot be made if all the necessary covariates are not supplied. Bayesian optimization for automatic tuning of the parameters of the algorithms was implemented as part of this work. However, it was found not to be necessary in many of the cases.

Comparison of the Algorithms:

In this paper, the metric used in comparing the algorithms is known as explained variance. The metric was calculated using Sci-Kit Learn. The metric is computed as follows:

explained variance = 1 – Var {Yact - Ypred}/Var{Yact}

where Ypred is the predicted parameter, Yact the measured parameter and Var is the variance (i.e. the square of standard deviation).

The maximum possible value is 1, representing perfect prediction; and lower values are worse. This metric is equivalent to the better-known R-squared metric for the case where the error has zero mean and is sometimes used interchangeably with R-squared. Like R-squared, this metric approximately mirrors the trend in the more common root mean square error metric. If the root mean square error score for an algorithm A is worse than that of algorithm B, the explained variance is very likely to follow the same trend.

Bubble Point Pressure Results:

Figs 1 – 7 below show the performance of the algorithms on the test data for bubble point pressure. Predicted values are on the y axis, while actual values are on the x-axis. In Fig. 2 PMF is plotted against the correlations of Standing and Vasquez-Beggs. The plot show that the PMF gives better performance than the two correlations. Also, there are instances where the two correlations break down either due to missing parameters or parameters outside the range of applicability of the correlations. The other machine learning techniques also show good prediction of the test data. Tuning of the hyper parameters was not required for any of the models. Table 1 shows a “League Table” of the performance of the methods in estimating bubble point pressure based on explained variance score.

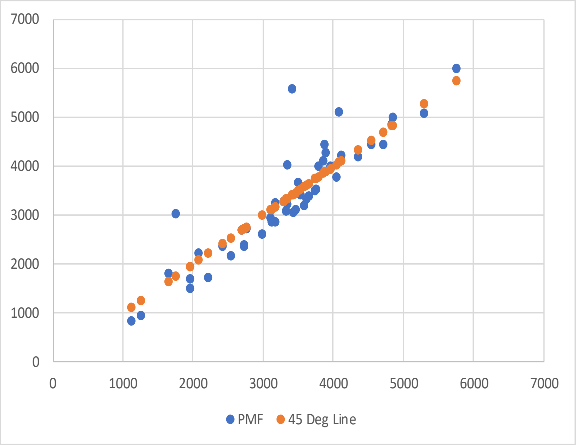

Fig1: PMF (Bubble Point Pressure, psia)

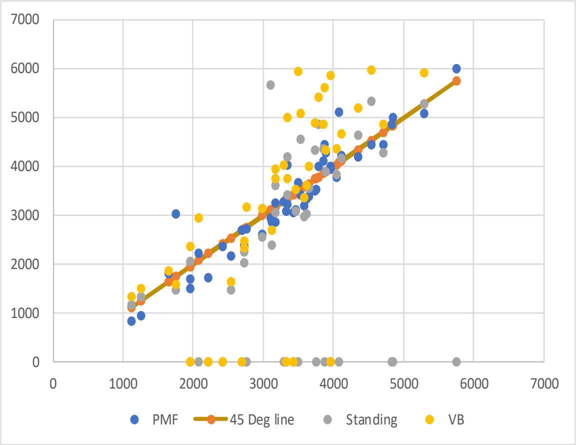

Fig 2: Cross Plot of BubP: Standing, VB and PMF

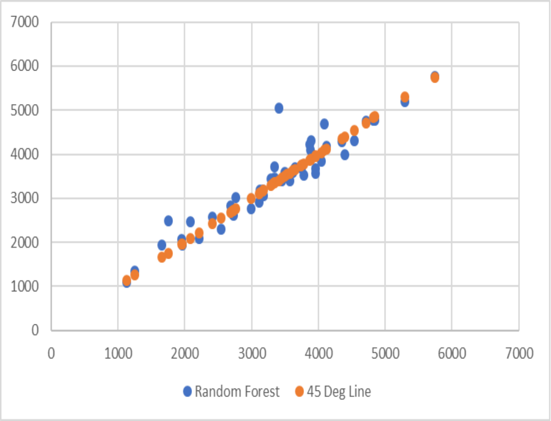

Fig 3: Random Forest (Bubble Point Pressure, psia)

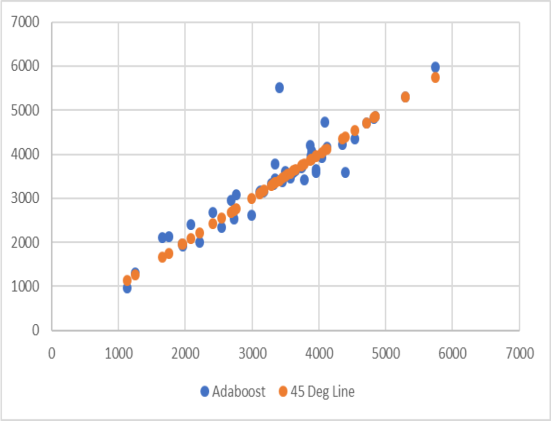

Fig 4: Adaboost (Bubble Point Pressure, psia)

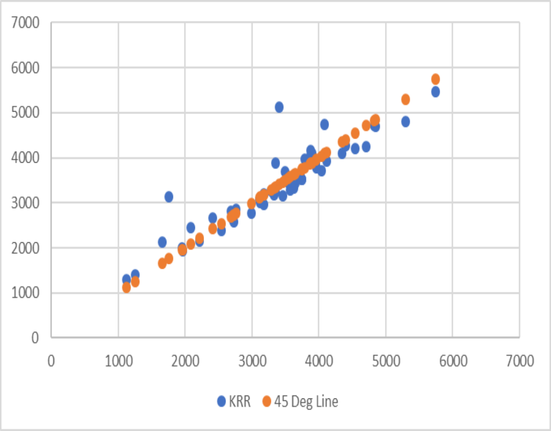

Fig 5: Kernel Ridge Regression (Bubble Point Pressure, psia)

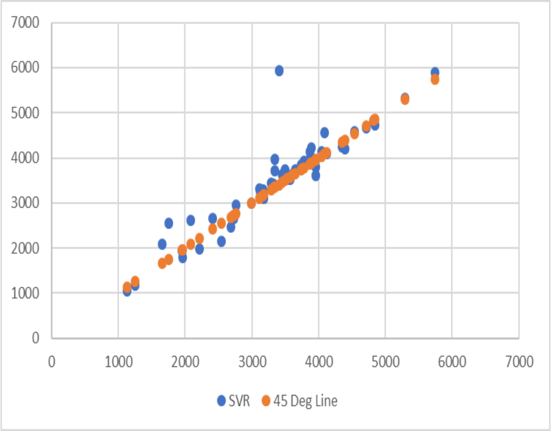

Fig 6: Support Vector Regression (Bubble Point Pressure, psia)

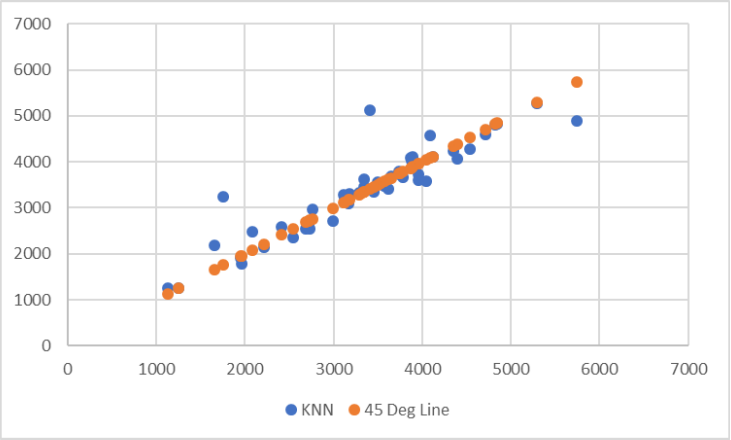

Fig 7: K Nearest Neighbors (Bubble Point Pressure, psia)

Bubble Point Prediction League Table:

Table 1 below give a ranking of the machine learning techniques and the correlations for bubble point pressure estimation based on explained variance score. The best performance is by Random Forest algorithm. A comparison of these ranking and the visual plots above provides some perspective on these scores. The relatively lower ranking of the Support Vector Regression is due to the influence of just one point it badly mis-estimated. If this point is taken out of reckoning the score for Support Vector Regression shoots to 0.95. The bottom-line therefore is that these machine learning algorithms perform very well in estimating the bubble point pressure and any of them could be used for this purpose and with the specified covariates. They perform better than the correlations - Standing and Vasquez-Beggs.

Table 1: Bubble Point Prediction League Table

Rank	Method	Score
1	Random Forest	0.893
2	Probabilistic Matrix Factorization	0.854
3	Adaboost	0.850
4	K Nearest Neighbours	0.839
5	Kernel Ridge Regression	0.838
6	Support Vector Regression	0.824
7	Standing Correlation	0.680
8	Vasquez-Beggs Correlation	0.560

Oil Formation Volume Factor:

Figs 8 – 14 below show the performance of the algorithms on the test data for oil formation volume factor. Predicted values are on the y axis, while actual values are on the x-axis. In Fig. 9 PMF is plotted against the correlations of Standing and Vasquez-Beggs. The figures show that all the machine learning methods and the correlations perform well for this parameter. This is also confirmed by the resulting league tables.

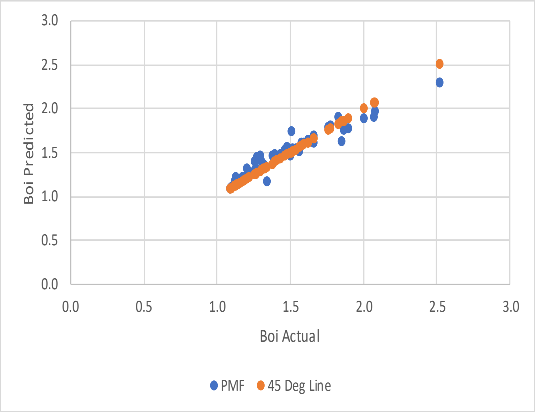

Fig 8: PMF (Oil Formation Volume Factor)

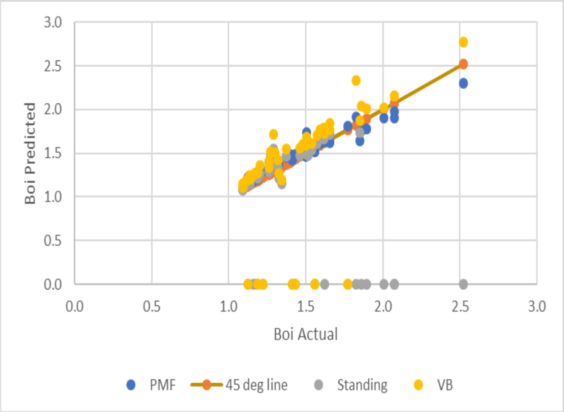

Fig 9: Cross Plot of Boi: Standing, VB and PMF

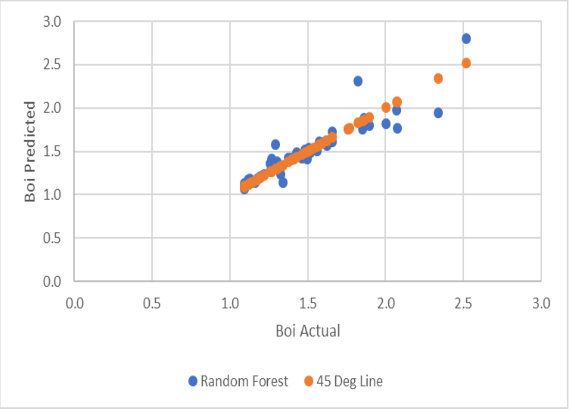

Fig 10: Random Forest (Oil Formation Volume Factor)

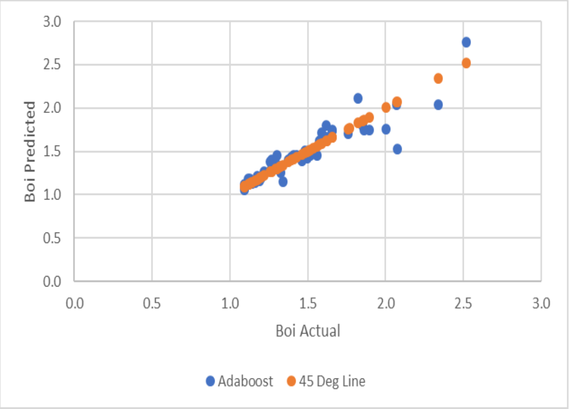

Fig 11: Adaboost (Oil Formation Volume Factor)

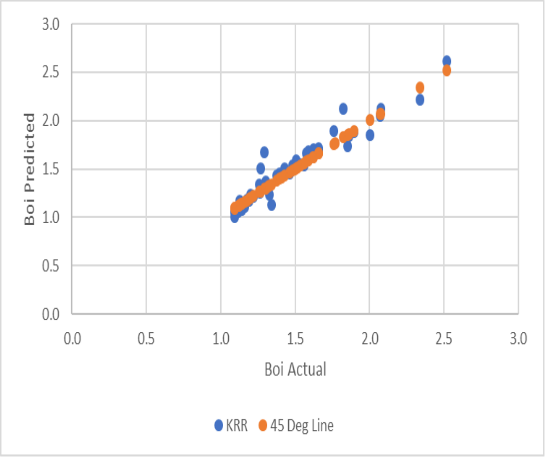

Fig 12: Kernel Ridge Regression (Oil Formation Vol. Factor)

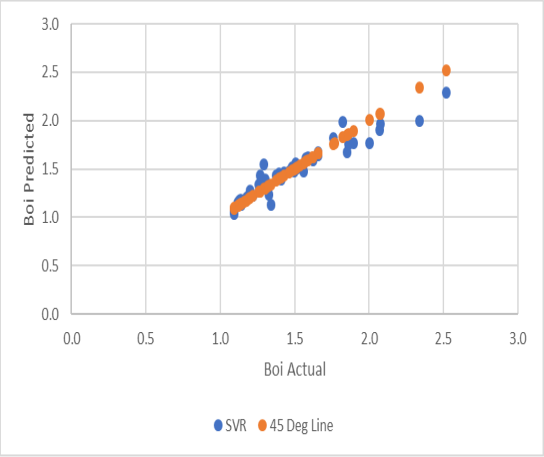

Fig 13: Support Vector Regression (Oil Formation Vol. Factor)

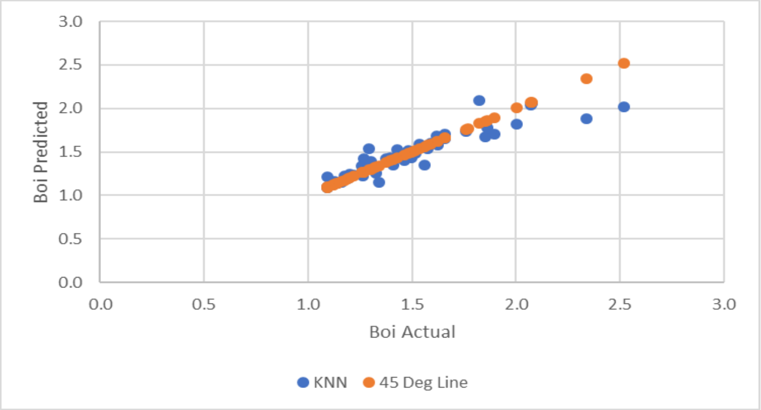

Fig 14: K Nearest Neighbors (Oil Formation Volume Factor)

Table 2: Oil Formation Volume Factor Prediction League Table

Rank	Method	Score
1	Probabilistic Matrix Factorization	0.910
2	Kernel Ridge Regression	0.908
3	Support Vector Regression	0.896
4	Standing Correlation	0.863
5	Vasquez-Beggs Correlation	0.846
6	Adaboost	0.845
7	Random Forest	0.840
8	K Nearest Neighbours	0.838

Oil Viscosity Results:

Figs 15 – 21 below show the performance of the algorithms on the test data for saturated oil viscosity. Predicted values are on the y axis, while actual values are on the x-axis. Majority of the data in the dataset are low viscosity data with viscosity less than 5 cp. In Fig. 16 PMF is plotted against the correlation Beggs and Robinson. The plot show that the PMF gives about the same performance as the correlation. The plot shown is for the case where dead oil viscosity is one of the covariates. All the methods except for Adaboost and Random Forest require dead oil viscosity to give significantly better performance.

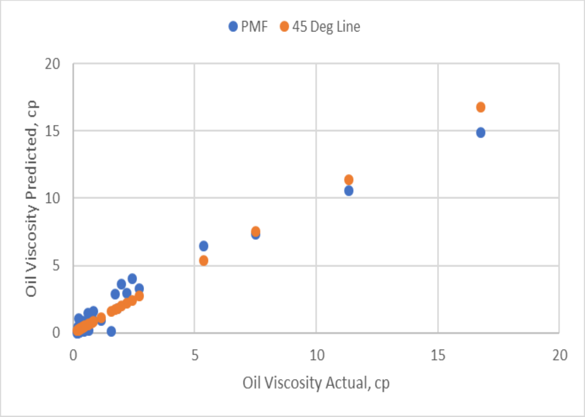

Fig 15: PMF (Oil Viscosity)

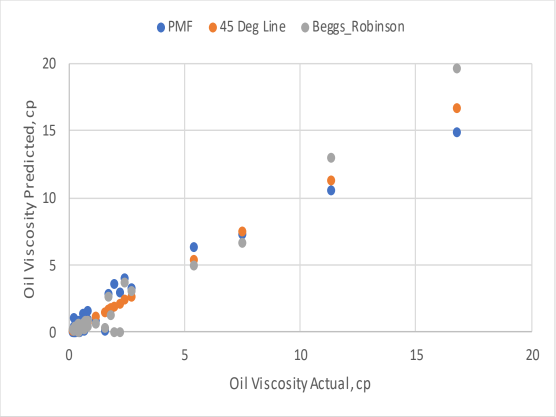

Fig 16: Cross Plot of Oil Viscosity, PMF and Beggs-Robinson

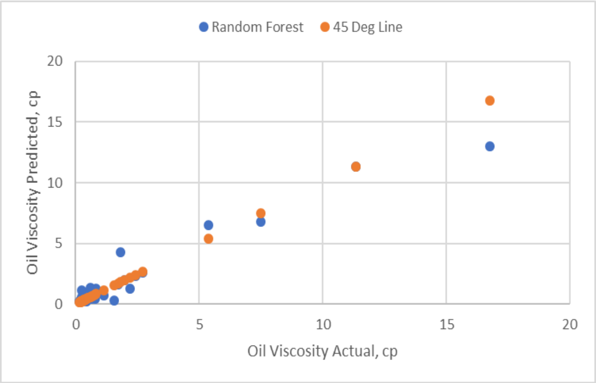

Fig 17: Random Forest (Oil Viscosity)

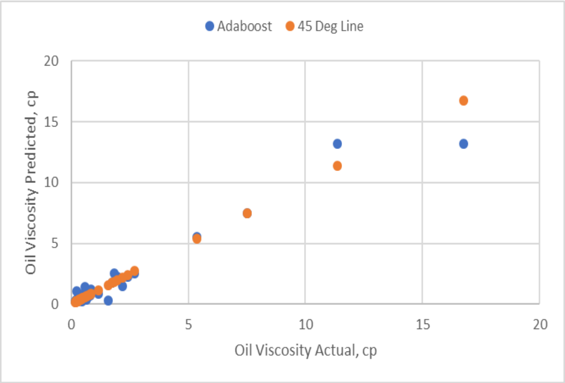

Fig 18: Adaboost (Oil Viscosity)

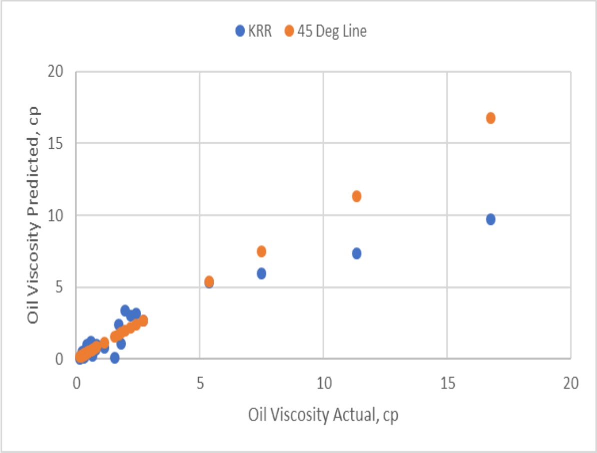

Fig 19: Kernel Ridge Regression (Oil Viscosity)

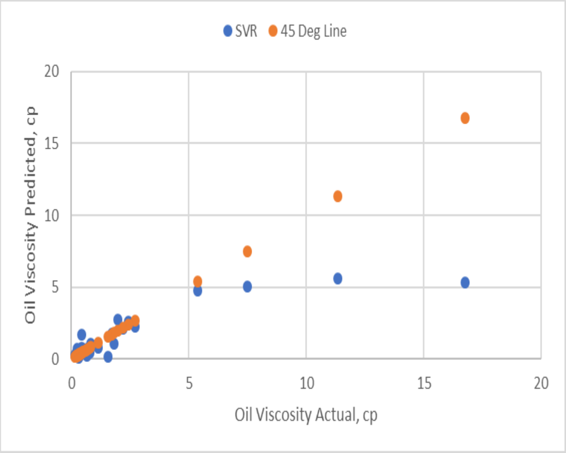

Fig 20: Support Vector Regression (Oil Viscosity)

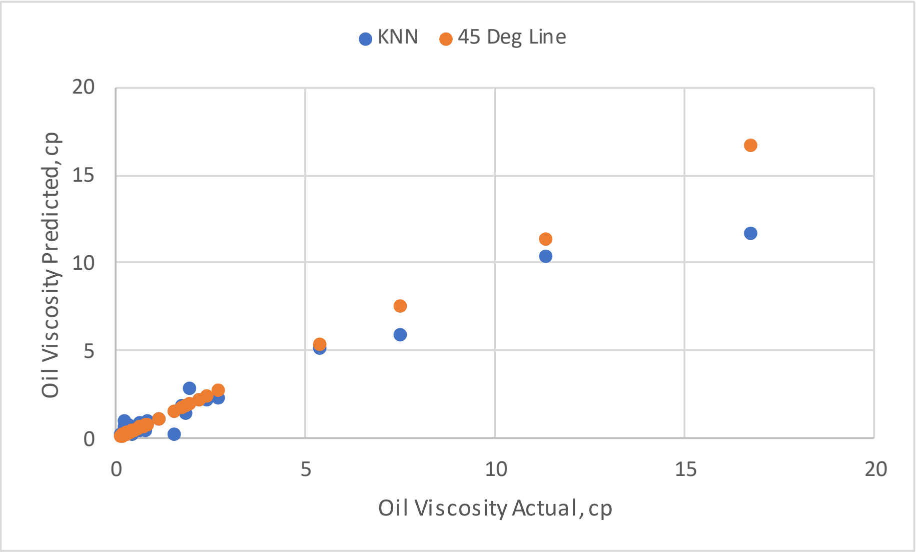

Fig 21: K Nearest Neighbors (Oil Viscosity)

Oil Viscosity Prediction League Table:

Table 3 below gives the ranking of the performance the algorithms in predicting oil viscosity in the test dataset. The ranking is based on the combined explained variance score achieved by the algorithms when dead oil viscosity data is included as part of the covariates and when it is excluded. The performance on oil viscosity varied depending on the type of algorithm. The ensemble- based techniques – Random Forest and Adaboost – gave very good performance without requiring dead oil viscosity data. The Kernel-based methods – Kernel Ridge Regression and Support Vector Regression – did not perform as well on this task with or without dead oil viscosity data. The results also provide insight into the modus operandi of the PMF technique. PMF seeks to find correlation in the features (or items); dead oil viscosity is highly correlated with live oil viscosity thus the PMF technique performs very well in this situation. The Beggs-Robinson correlation outperforms most of the machine learning techniques when dead oil data is present. The implication of this is that when dead oil viscosity and initial solution gas oil ratio data are available, the simple Beggs-Robinsons correlation will be the surer bet to use for live oil viscosity estimation; of course, within the limits of its applicability.

Table 3: Oil Viscosity Prediction League Table

Rank	Method	Score with Dead Oil Visc	Score Without Dead Oil Visc	Total
1	Adaboost	0.952	0.943	1.895
2	Random Forest	0.938	0.912	1.850
3	K Nearest Neighbours	0.924	0.752	1.676
4	Probabilistic Matrix Factorization	0.959	0.584	1.543
5	Kernel Ridge Regression	0.826	0.410	1.236
6	Support Vector Regression	0.611	0.259	0.870
	Beggs Robinson	0.960	NA	NA

Application to Gas PVT

The gas dataset used in this study is very limited, consisting of only 72 data points. This reflects the fact that gas development in the Niger Delta only picked up steam as from the late 1990's. The parameters that were to be predicted after the machine learning were dew point pressure, gas expansion factor and condensate gas ratio. The fitting parameters were depth, initial reservoir pressure, reservoir temperature, gas compressibility factor (Z factor) and gas gravity. When dew point pressure and gas expansion factor were to be predicted, condensate gas ratio was included as a fitting parameter and for condensate gas ratio prediction, dew point pressure and gas expansion factor were included as fitting parameters. For gas expansion factor, the Z factor was excluded.

Dew Point Pressure Results:

The methods did not perform as well in predicting dew point pressure compared to bubble point pressure. Figs 22 – 25 show the performance for PMF, Adaboost, Kernel Ridge Regression and Support Vector Regression techniques. Adaboost gave the best performance. Table 4 gives the explained variance scores for all the techniques. The poor scores for PMF and KRR are due to the limited data and the inability of the methods to predict the dew points of two highly undersaturated reservoirs. The better score recorded for Support Vector Regression was obtained after tuning the hyper-parameters using the Bayesian Optimization program GpyOpt. The was the only case in this work where auto-tuning of the hyper-parameters resulted in a significant improvement. The fact that optimizing independently on the training data for Support Vector Regression resulted in improved prediction of the test data indicate that the two highly undersaturated cases are not outliers.

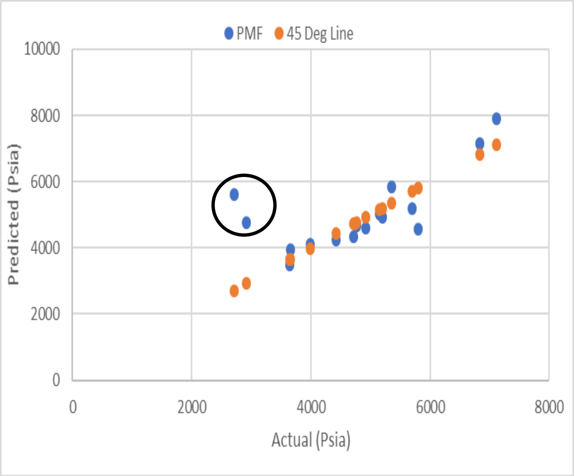

Fig 22: PMF (Dew Point Pressure)

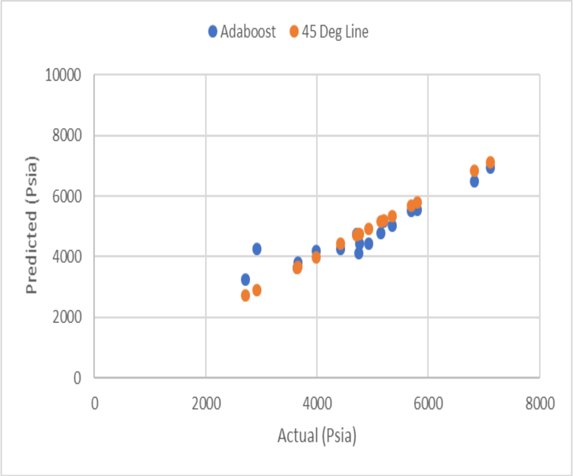

Fig 23: Adaboost (Dew Point Pressure)

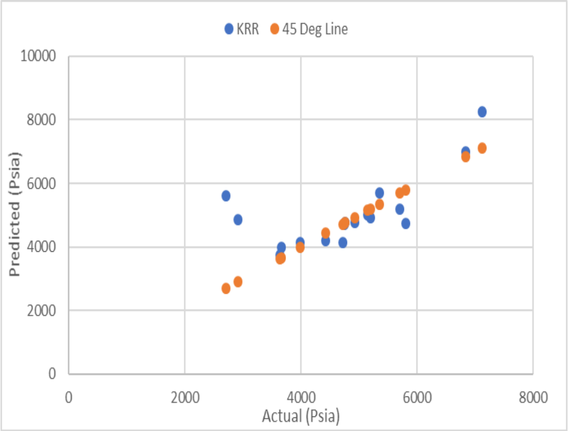

Fig 24: Kernel Ridge Regression (Dew Point Pressure)

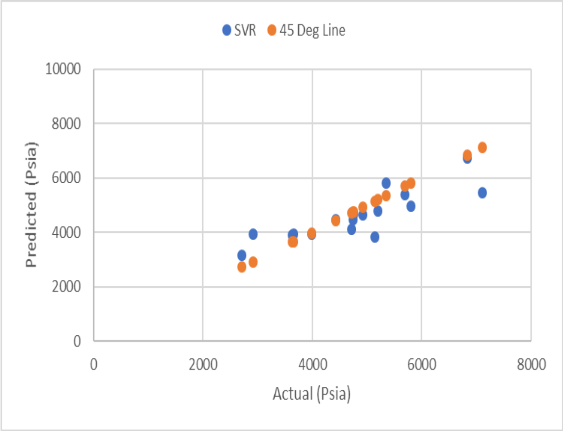

Fig 25: Support Vector Regression (Dew Point Pressure)

Table 4: Dew Point Pressure Prediction League Table

Rank	Method	Score
1	Adaboost	0.855
2	Support Vector Regression	0.703
3	Random Forest	0.681
4	K Nearest Neighbours	0.679
	PMF	0.382
	Kernel Ridge Regression	0.373

Gas Expansion Factor Results:

If Pressure, Temperature and Z factor data for a reservoir are available, gas expansion factor can be directly calculated from first principles. The interesting machine learning problem therefore will be to attempt to estimate gas expansion factor from other fluid data when the Z factor is not available. In other words, are we able to unravel the latent effect of the Z factor using machine learning techniques from other available fluid data (e.g. gas gravity, depth, pressure, temperature, CGR)? This was carried out by excluding the Z factor from the covariates used in training. The methods did not replicate the excellent performance obtained for oil formation volume factor. Good estimates were obtained using the ensemble techniques -Adaboost and Random Forest – though with low explained variance score of approximately 0.6, and Mean Absolute Error(MAE) of 5%. K nearest neighbor algorithm also performed well with MAE of 5%, but with low explained variance score of 0.418. The low explained variance is due to the consistent misfit (though low) seen in the prediction for most of the data points (see Fig. 26 and 27). Figs 28 and 29 show the adaboost prediction plotted with estimates of the gas expansion factor obtained after estimating the Z factor using the correlations of Dranchuk and Abou-Kassem and that of Brill and Beggs respectively. The plots show that these conventional techniques perform as well or even better than the Adaboost algorithm in predicting the gas expansion factor for most of the data, except for two prominent misfits that occur at high pseudo-reduced pressures ($P_{pr} > 10$). It may well be that it is at higher pseudo-reduced pressures that these machine learning techniques will add value to gas formation volume factor estimation. However, training and testing with more datasets is necessary before a definitive conclusion can be drawn.

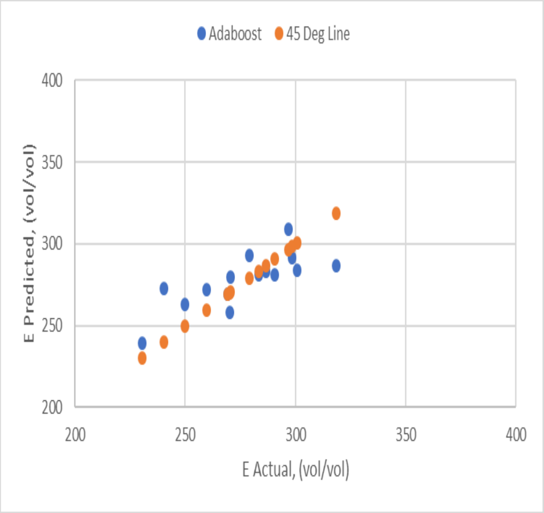

Fig 26: Adaboost Gas Expansion Factor)

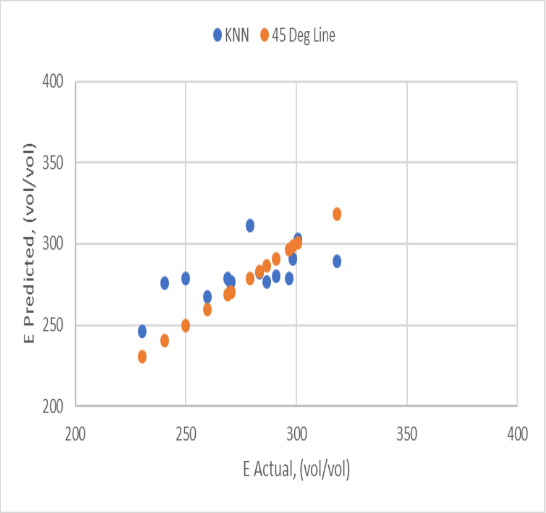

Fig 27: KNN (Gas Expansion Factor)

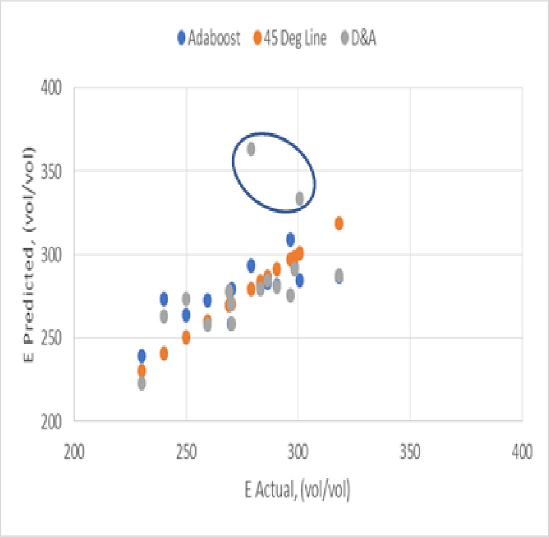

Fig 28: Dranchuk & Abou-Kassem(for Z) (Gas Expansion Factor)

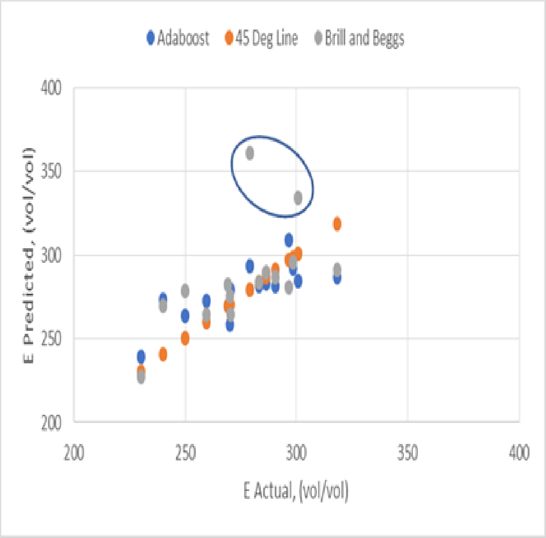

Fig 29: Brill and Beggs (for Z) (Gas Expansion factor)

Condensate Gas Ratio Results:

Condensate Gas Ratio(CGR) estimation is very important in gas development. Proper estimation of associated condensate volumes is required for facility sizing. Also, associated condensates tend to improve the value of gas projects. The machine learning algorithms of this work were tested for their ability to predict condensate volumes on test data after training. It was found that the Kernel-based algorithms – Kernel Ridge Regression and Support Vector Regression - gave reasonably good prediction of the test CGR data with explained variance scores of 0.776 and 0.802 respectively. The other algorithms gave poor results with explained variance scores less than 0.3. The fact that the Kernel-based methods singularly gave better results could be pointing to something -namely that CGR prediction may be favoured by projecting the features to higher dimensions. This theory if true could indicate future directions for perfecting machine learning techniques for CGR prediction or for improving conventional empirical/semi-empirical CGR prediction correlations. Figs 30 and 31 show the performance of the Kernel-based algorithms.

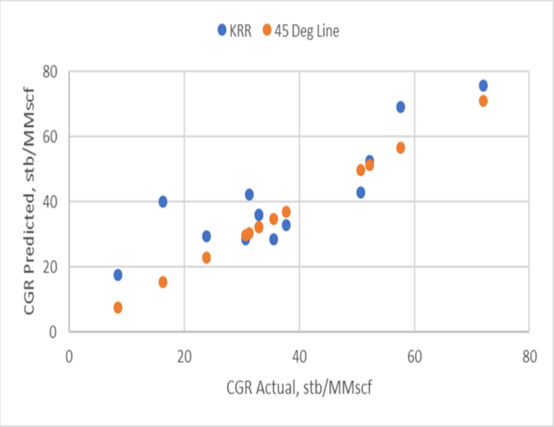

Fig 30: Kernel Ridge Regression (Condensate Gas Ratio)

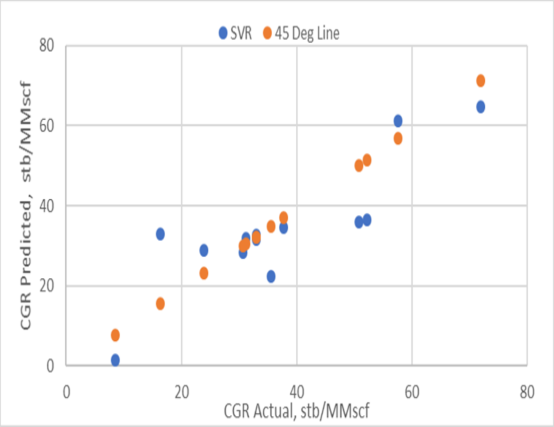

Fig 31: Support Vector Regression (Condensate Gas Ratio)

Conclusion

Machine Learning techniques – Collaborative Filtering (Probabilistic Matrix Factorization), Random Forest, Adaptive Boosting(Adaboost), Kernel Ridge Regression, Support Vector Regression and K Nearest Neighbors – were used to train and test oil and gas PVT datasets from the Niger Delta. The methods showed great promise in the prediction of oil PVT parameters - Bubble Point Pressure, Oil Formation Volume Factor and Saturated Oil Viscosity. The ensemble-based techniques -Adaboost and Random Forest - when trained can predict Saturated Oil Viscosity without requiring Dead Oil Viscosity data. The Machine Learning techniques evaluated did not perform as well on the limited gas PVT data. Predictions of Dew Point Pressure were reasonable, while the Kernel-based techniques – Kernel Ridge Regression and Support Vector Regression – gave good predictions of the Condensate Gas Ratio – a parameter for which the other techniques gave poor prediction.

One interesting finding from the study is that Collaborative Filtering which was developed for consumer products recommendation systems can be applied to an engineering problem. Further tests of all the techniques with other datasets, especially larger gas datasets from the Niger Delta and other basins are necessary to determine whether the findings of this work represent a general trend or are dataset-dependent. It will also be interesting to find out if more sophisticated Machine Learning techniques, such as Deep Learning, offer any significant advantage over the techniques considered in this work for the fluid properties estimation problem.

References

1. Mohaghegh , S. D.,Arefi, R and Ameri, S.: “A Methodological Approach for Reservoir Heterogeneity Characterisation Using Artificial Neural Networks”. SPE 28394, 1994
2. Wong, P. M., Henderson, D. J. and Brooks, L. J.: “Permeability Determination Using Neural Networks in the Ravva Field, Offshore India”. SPE Reservoir Evaluation & Engineering, pg. 99 – 104, April 1998.
3. Badarinah, V., Suryanarayana, K., Fahd Zaki, Y., Khalid, S., and Antonio, V.: “Log-Derived Permeability in a Hetergenous Carbonate Reservoir of Middle East, Abu Dhabi, Using Artificial Neural Networks”. SPE 74345. Presented at the SPE International Conference and Exhibition, Villahermosa, Mexico, Feb. 2002.
4. Tian, C. and Horne, R, N..: “Applying Machine Learning Techniques to Interprete Flow Rate, Pressure and Temperature Data from Permanent Downhole Gauges”. SPE-174034-MS, Presented at the SPE Western Regional Meeting, Garden Grove, California, U.S.A, April 2015.
5. Mohaghegh, S. D., Hafez, H., Gaskari, R., Haajizadeh, M. and Kenawy, M.: “Uncertainty Analysis of a Giant Oil Field in the Middle East Using Surrogate Reservoir Model”. SPE 101474, Proceedings of the International Petroleum Exhibition and Conference, Abu Dhabi, UAE, Nov 2006.

6. Mohaghegh, S. D., Liu, J., Gaskari, R., Maysami M. and Olukoko, G.: "Application of Surrogate Reservoir Models(SRMs) to an Onshore Green Field in Saudi Arabia, Case Study". SPE 151994. Presented at the North Africa Technical Conference and Exhibition, Cairo, Egypt, February 2012.
7. Mohaghegh, S. D., Liu, J., Gaskari, R., Maysami M. and Olukoko, G.: "Application of Well-Based Surrogate Reservoir Models(SRMs) to two Offshore Fields in Saudi Arabia, Case Study". SPE 153845. Presented at the SPE Western North American Regional Meeting, Bakersfield California, U.S.A., March 2012.
8. Gomez, Y., Khazemi, Y. and Mohagegh, S. D.: "Top Down Intelligent Reservoir Modelling (TDIRM)". SPE 124204. Presented at the SPE Annual Technical Conference and Exhibition. New Orleans Louisiana, U.S.A., Oct 2009.
9. Moussa, T., Elkatny, S., Abdulraheem, A., Mahmaid, M. and Alloush, R.: "A hybrid Artificial Intelligence Method to Predict Gas Solubility and Bubble Point Pressure". SPE 188102, Presented at the SPE Kingdom of Saudi Arabia Annual Technical Symposium, Damman, Saudi Arabia June 2017
10. Salakhutdinov, R. and Mnih, A.: "Advances in Neural Information Processing Systems". Vol. 20, pg. 1257 – 1264, 2008.