



**NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF PETROLEUM AND NATURAL GAS
ENGINEERING**

**ANALYSIS AND PERFORMANCE PREDICTION OF SHALE WELLS USING
DATA ANALYTICS AND MACHINE LEARNING**

M.Sc. THESIS

John Kaninda MFUTA

**Nicosia
June 2023**

JOHN KANINDA MFUTA

**ANALYSIS AND PERFORMANCE PREDICTION OF SHALE WELLS
USING DATA ANALYTICS AND MACHINE LEARNING**

MASTER THESIS 2023

**NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF PETROLEUM AND NATURAL GAS
ENGINEERING**

**ANALYSIS AND PERFORMANCE PREDICTION OF SHALE WELLS USING
DATA ANALYTICS AND MACHINE LEARNING**

M.Sc. THESIS

John Kaninda MFUTA

Supervisor

Prof. Dr. Cavit ATALAR

Co-Supervisor

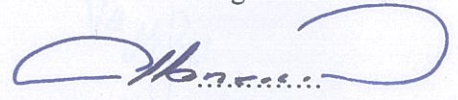
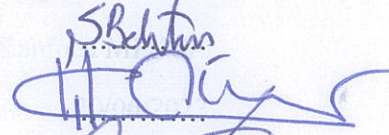
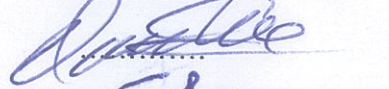


Assoc. Prof. Dr. Emre ARTUN

Nicosia

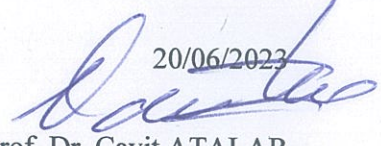
June 2023

Approval

We certify that we have read the thesis submitted by John Kaninda MFUTA, titled “**Analysis and Performance Prediction of Shale Wells Using Data Analytics and Machine Learning**” and that in our combined opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Applied Sciences.

Examining Committee	Name-Surname	Signature
Head of the Committee:	Prof. Dr. Salih SANER	
Committee Member:	Prof. Dr. Şenol BEKTAŞ	
Committee Member:	Assoc. Prof. Dr. Hüseyin ÇAMUR	
Supervisor:	Prof. Dr. Cavit ATALAR	
Co-Supervisor:	Assoc. Prof. Dr. Emre ARTUN	

Approved by the Head of the Department

20/06/2023

Prof. Dr. Cavit ATALAR
Head of Department

Approved by the Institute of Graduate Studies

20/06/2023
Prof. Dr. Kemal Hüsni Can Başer

Head of the Institute



Declaration

I hereby declare that all information, documents, analysis, and results in this thesis have been collected and presented according to the academic rules and ethical guidelines of the Institute of Graduate Studies, Near East University. I also declare that as required by these rules and conduct, I have fully cited and referenced information and data that are not original to this study.

John Kaninda MFUTA

20/06/2023

Acknowledgments

I want to convey my profound gratitude to my supervisor Prof. Dr. Cavit Atalar, who oversaw my research. He helped me get accepted to The Near East University, which was a significant turning point in my life. His advice is crucial to my academic development, expertise, and scientific thinking.

I also want to convey my sincere appreciation and admiration to my co-supervisor, Assoc. Prof. Dr. Emre Artun, who has given me a lot of support and consistently supplied insight throughout our sessions and this thesis. His dynamism, vision, passion, and drive have profoundly moved me. I also want to thank Prof. Dr. Salih Saner, Assoc. Prof. Dr. Serhat Canbolat, and all of my lecturers who helped me with my master's degree both in-person and online. I am truly grateful for the knowledge you all shared with me.

My parents, Angelique Misenga Kayembe and Jean Mari Kaninda Mukumbi, deserve a special word of appreciation. In appreciation for their love, support, and inspiration in helping me to finish this thesis, I'd want to thank my brothers, sisters, cousins, and nephews. And a special dedication to Dieu Donné Ndibu Kaninda.

Additionally, I want to express my gratitude to my friends and classmates for their support and suggestions throughout my study period, both in class and at home.

John Kaninda MFUTA

Abstract

Analysis and Performance Prediction of Shale Wells Using Data Analytics and Machine Learning

MFUTA, John Kaninda

M.Sc. Department of Petroleum and Natural Gas Engineering

June, 2023, 93 pages

The oil and gas industry, in particular, has benefited greatly from the ability of data analytics to improve operations and save time. Numerical reservoir simulators and laboratory tests are currently employed for modeling. However, the computational cost of these approaches is substantial. In addition, research on the application of data-driven statistical methodologies has been conducted, with a primary focus on analysis and performance prediction in unconventional reservoirs. An analysis of the key factors that influence the cumulative gas produced after 1-year MCF from unconventional reservoirs was done in this study using data analytics to gain new knowledge.

Predictive modeling and exploratory data analysis were the main methods employed. Analyzing exploratory data indicated a connection between the operational and reservoir parameters. Additionally, algorithms based on statistics and machine learning were created to forecast the total amount of gas produced after one-year MCF. These predictive models were compared, and it was found that random forest had the lowest prediction error and highest R^2 value, making it the preferable approach.

Finally, variable importance was used to identify the performance prediction criteria in unconventional shale gas reservoirs that had the most influence. It is interesting to note that the factors that have the biggest effects on the stimulated reservoir volume. Our results indicate that operational parameter is more essential than reservoir parameter in stimulating reservoir volume and driving high performance, with Bottom Perf(ft) and Cluster per stage being the most crucial factors in achieving high performance.

Keywords: Exploratory data analysis, Predictive modeling, Unconventional reservoirs

Özet

Veri Analitiği ve Makine Öğrenimi Kullanılarak Şeyl Kuyularının Analizi ve Performans Tahmini

MFUTA, John Kanında

M.Sc. Petrol ve Doğal Gaz Mühendisliği Bölümü

Haziran, 2023, 93 sayfa

Özellikle petrol ve gaz endüstrisi, veri analitiğinin operasyonları iyileştirme ve zamandan tasarruf etme yeteneğinden büyük ölçüde faydalandı. Modelleme için şu anda sayısal rezervuar simülatörleri ve laboratuvar testleri kullanılmaktadır. Bununla birlikte, bu yaklaşımların hesaplama maliyeti önemlidir. Ek olarak, geleneksel olmayan rezervuarlarda analiz ve performans tahminlerine odaklanarak, veriye dayalı istatistiksel metodolojilerin uygulanmasına ilişkin araştırmalar yürütülmüştür. Bu çalışmada, yeni bilgiler elde etmek için veri analitiği kullanılarak geleneksel olmayan rezervuarlardan 1 yıllık MCF'den sonra üretilen kümülatif gazı etkileyen temel faktörlerin analizi yapılmıştır.

Tahmine dayalı modelleme ve keşifsel veri analizi, kullanılan ana yöntemlerdi. Keşif verilerinin analizi, operasyonel ve rezervuar parametreleri arasında bir bağlantı olduğunu gösterdi. Ek olarak, bir yıllık MCF'den sonra üretilen toplam gaz miktarını tahmin etmek için istatistik ve makine öğrenimine dayalı algoritmalar oluşturuldu. Bu tahmin modelleri karşılaştırıldı ve rastgele ormanın en düşük tahmin hatasına ve en yüksek R^2 değerine sahip olduğu ve onu tercih edilen bir yaklaşım haline getirdiği bulundu.

Son olarak, geleneksel olmayan kaya gazı rezervuarlarında en fazla etkiye sahip olan performans tahmin kriterlerini belirlemek için değişken önem kullanıldı. Uyarılmış rezervuar hacmi üzerinde en büyük etkiye sahip olan faktörlerin dikkate alınması ilginçtir. Bulgularımız, operasyonel parametrenin, rezervuar hacmini canlandırmada ve yüksek performans sağlamada rezervuar parametresinden daha önemli olduğunu, yüksek performans elde etmede en önemli faktörlerin aşama başına Alt Perf(ft) ve Küme ile birlikte olduğunu göstermektedir.

Anahtar Kelimeler: Keşifsel veri analizi, Tahmine dayalı modelleme, Geleneksel olmayan rezervuarlar

Table of Contents

Approval	2
Declaration.....	3
Acknowledgments	4
Abstract	5
Özet.....	6

CHAPTER I

Introduction.....	15
Statement of the Problem	16
Purpose of the Study	16
Hypothesis.....	16
Objectives and Questions	16
Significance of the Study	17
Limitations	17
Definition of Terms.....	17
Summary of the Study.....	18

CHAPTER II

Literature Review	19
Reservoir Modeling.....	19
Exploratory Data Analysis	20
Unconventional Gas Recovery.....	22
Horizontal Drilling	23
Hydraulic Fracturing	24
<i>Types of Hydraulic Fracturing</i>	24
Data Mining.....	25
Artificial Intelligence	25
Machine Learning	25

Programming Languages.....	26
Software and Frameworks.....	27
Related Research.....	27
General workflow.....	28

CHAPTER III

Research Methodology	29
Research Design.....	29
Population of the Study.....	29
Validity and Reliability Criteria.....	29
Method of Data Collection.....	30
Data Analysis Technique (Exploratory Data Analysis of Data Set).....	30
Machine Learning (Supervised Learning Algorithm).....	30
<i>Metrics for Evaluating Regression</i>	31
Tree Methods.....	32
<i>Decision Tree</i>	32
<i>Random Forest</i>	35
Implementation of Decision Tree and Random Forest Using Scikit-learn.....	36

CHAPTER IV

Results and Discussion	37
Descriptive Statistics.....	37
<i>Reservoir and Operational Parameters</i>	37
Univariate Data Analysis	40
<i>Boxplots for Reservoir and Operational Parameters</i>	40
<i>Histograms for Reservoir and Operational Parameters</i>	45
<i>Scatterplot for Reservoir and Operational Parameters</i>	50
Multivariate Correlation Plot	55
Predictive modeling.....	58
<i>Decision Tree</i>	58
<i>Random Forest</i>	64

Comparison of Decision Tree and Random Forest Using Bar Chart Plot	71
Variable Importance	72
Comparison of Feature Importance Rankings of DT and RF	73

CHAPTER V

Concluding Remarks.....	74
Conclusions	74
Recommendations	75
References.....	76
Appendices.....	78
Appendix A Import Relevant packages	79
Appendix B Decision Tree Classifier Building in Scikit-learn.....	80
Appendix C Random Forest implementation using scikit-learn	85
Appendix D Variables of Multivariable Correlation Plot for Reservoir Parameters	89
Appendix E Variables of Multivariable Correlation Plot for Operational Parameters	90
Appendix F Similarity Report.....	91
Appendix G Ethical Approval Letter	92

List of Tables

Table 4.1 Descriptive statistics for reservoir and operational parameters.....	39
Table 4.2 Training and testing sets for decision tree.....	59
Table 4.3 Training and testing sets for random forest.....	65

List of Figures

Figure 2.1. Scatterplot matrix (Schuetter et al. 2018)	20
Figure 2.2. Histograms for predictor variables (Zhong et al., 2015)	21
Figure 2.3. Scatterplot matrix for predictor variables (Zhong et al., 2015)	21
Figure 2.4. An illustration of shale gas compared to other types of gas deposits. (Stevens, Paul August 2012).	23
Figure 2.5. Hydraulic fracturing process (introduction to well testing. schlumbecath, bath, england,1998)	24
Figure 2.6. Workflow for data analytics approach	28
Figure 3.1. Decision tree illustration. (Breiman, L. 2011)	33
Figure 3.2. Decision tree versus random forest	35
Figure 4.1. Reservoir parameters box plots: (a) Initial Pressure Estimate (psi), (b) Reservoir Temperature(F), (c) Net Pay(ft), (d) Porosity, (e)Water Saturation, (f) Oil saturation, (g) Gas Saturation, (h) Gas Specific Gravity, (j) CO2 (k)N2	42
Figure 4.2. Operational parameters box plots: (a) TVD,(b)Spacing ,(c) Stages ,(d) Number of clusters , (e) Number of clusters per Stage , (f) Total Proppant (MM Lbs.), (g) Lateral Length (ft.), (h) Top Perf (ft.), (i) Bottom Perf (ft.), (j) Sand face Temp (F), (k) Static wellhead Temp (F), (l) Cumulative Gas Produced after 1 year, MCF	44
Figure 4.3. Reservoir parameters histogram plots: (a) Initial Pressure Estimate (psi), (b) Reservoir Temperature(F), (c) Net Pay(ft), (d) Porosity, (e)Water Saturation, (f) Oil saturation, (g) Gas Saturation, (h) Gas Specific Gravity, (j) CO2 (k) N2	47
Figure 4.4. Operational parameters histogram plots: (a) TVD,(b)Spacing ,(c) Stages , (d) Number of clusters , (e) Number of clusters per Stage , (f) Total Proppant (MM Lbs.), (g) Lateral Length (ft.), (h) Top Perf (ft.), (i) Bottom Perf (ft.), (j) Sand face Temp (F), (k) Static wellhead Temp (F), (l) Cumulative Gas Produced after 1 year, MCF	49

- Figure 4.5.** Reservoir parameters scatter plots: (a) Initial Pressure Estimate (psi), (b) Reservoir Temperature(F), (c) Net Pay(ft), (d) Porosity, (e)Water Saturation, (f) Oil saturation, (g) Gas Saturation, (h) Gas Specific Gravity, (j) CO2 (k) N2 52
- Figure 4.6.** Operational parameters scatter plots: (a) TVD,(b)Spacing ,(c) Stages , (d) Number of clusters, (e) Number of clusters per Stage , (f) Total Proppant (MM Lbs.), (g) Lateral Length (ft.), (h) Top Perf (ft.), (i) Bottom Perf (ft.), (j) Sand face Temp (F), (k) Static wellhead Temp (F), (l) Cumulative Gas Produced after 1 year, MCF 54
- Figure 4.7.** Multivariable correlation plot for reservoir parameters: (a) Initial Pressure Estimate (psi), (b) Reservoir Temperature(F), (c) Net Pay(ft), (d) Porosity, (e)Water Saturation, (f) Oil saturation, (g) Gas Saturation, (h) Gas Specific Gravity, (j) CO2 (k) N2 56
- Figure 4.8.** Multivariable correlation plot for operational parameters: (a) TVD,(b)Spacing, (c) Stages ,(d) Number of clusters , (e) Number of clusters per Stage , (f) Total Proppant (MM Lbs.), (g) Lateral Length (ft.), (h) Top Perf (ft.), (i) Bottom Perf (ft.), (j) Sand face Temp (F), (k) Static wellhead Temp (F), (l) Cumulative Gas Produced after 1 year, MCF 57
- Figure 4.9.** Training actual vs prediction using decision tree (70% training) 59
- Figure 4.10.** Testing actual vs prediction using decision tree (30% training) 60
- Figure 4.11.** Feature importance score using decision tree (70/30) 60
- Figure 4.12.** Training actual vs prediction using decision tree (75% training) 61
- Figure 4.13.** Testing actual vs prediction using decision tree (25% training) 61
- Figure 4.14.** Feature importance score using decision tree (75/25) 62
- Figure 4.15.** Training actual vs prediction using decision tree (80% training) 63
- Figure 4.16.** Testing actual vs prediction using decision tree (20% training) 63
- Figure 4.17.** Feature importance score using decision tree (80/20) 64
- Figure 4.18.** Training actual vs prediction using random forest (70% training) 66

Figure 4.19. Testing actual vs prediction using random forest (30% training)	66
Figure 4.20. Feature importance score using random forest (70/30)	67
Figure 4.21. Training actual vs prediction using random forest (75% training)	68
Figure 4.22. Testing actual vs prediction using random forest (25% training)	68
Figure 4.23. Feature importance score using random forest (75/25)	69
Figure 4.24. Training actual vs prediction using random forest (80% training)	70
Figure 4.25. Testing actual vs prediction using random forest (20% training)	70
Figure 4.26. Feature importance score using random forest (80/20)	71
Figure 4.27. Decision tree and random forest bar chart comparison of training and testing tests	72

List of Abbreviation

AI	Artificial intelligence
ANN	Artificial neural network
DCA	Decline curve analysis
DT	Decision tree
EUR	Estimated ultimate recovery
Eq	Equation
GBRT	Gradient boosting for regression tree
GIP	Gas in place
GLR	Gas-liquid ratio
HC	Hydrocarbon
HF	Hydraulic fracture
LR	Linear regression
ML	Machine learning
MAE	Mean absolute error
MCF	Thousand cubic feet
MSE	Mean square error
PLS	Partial least square
R^2	Coefficient of determination
RF	Random forest
RMSE	Root mean square error
SL	Supervised learning
SVM	Support vector machine
TOC	Total organic content
TVD	Total vertical depth

CHAPTER I

Introduction

Shale gas, tight shales, gas hydrates, and coal bed methane production are encouraged by rising demand for fossil fuels and a decrease in their production. Gas that occurs in these unconventional reservoirs is now producible and has enormous potential for use in the future due to advancements in drilling and production technologies.

Unconventional natural gas known as shale gas can be found trapped in shale strata. (Ertekin,200) Shale gas has become more affordable to extract in large quantities during the 1990s because to a combination of horizontal drilling and hydraulic fracturing, and some analysts predict that shale gas will significantly increase the world's energy supply. (Clifford Krauss,2009).

Although shale gas reservoirs have been known for many decades, producing gas from these unconventional reservoirs started as the demand for natural gas increased, the production become commercially feasible, and the availability of drilling and completion technologies become implementable in late 1850s. (Paul, 2012)

The key technology in shale gas production is hydraulic fracturing. For commercially feasible gas production, shale gas reservoirs need to be hydraulically fractured. Hydraulic fracturing and horizontal well technology became operational for oil industry in 1980s to produce oil or gas from shale gas reservoirs. These fractures increase contact surface areas with production zone resulting in an increase in well productivity. (Krauss, Clifford,2009).

However, massive body of academic research building on top of each other over the years attempted to provide solutions to these problems-based formation type and economic feasibility of extracting shale gas. This has gained development in simulation study and data mining to get insight into the viability of shale well. This methodology presents complex computations that can only be achieved with sophisticated systems and some commercial simulation software (Ludmilla 2018). Machine learning techniques were also applied to effectively analyses reservoir recovery and develop an intuitive model for predictive analytics (Chen,2022). This research study aims to analyze and develop a machine learning model that evaluates the performance prediction of shale

well and serves as a useful tool for predicting cumulative gas produced after 1-year MCF and recovery factors.

Statement of the Problem

Global energy demand is on the rise to power machinery and the oil and gas industries are faced with a big task to meet with such rising demands. Recovery of oil and gas is a major concern to the industry and heavy investment have been made to explore different techniques to recover more gas. Horizontal drilling and hydraulic fracturing such technique widely used as it enables to improve recovery of shale well.

Therefore, prediction of dependent variable becomes a useful tool to give an insight towards making more practical decisions which has seen many theoretical models fail to deliver a more precise pattern. Machine learning is a data mining technique that will be explored in this work to create a powerful predictive tool.

Purpose of the Study

The purpose of this study is to analyze and explore deeper into data mining techniques as applied in shale well reservoirs and measure its performance by comparing two different machine learning models to predict cumulative gas produced after 1-year MCF.

Hypothesis

If a machine learning model is sufficiently trained, it will produce accurate results for a variety of instances. Within the parameters for which it was trained, it discovers the relationship between the input and the output and can eventually predict (with a certain level of accuracy) the output for a given input.

Objectives and Questions

A sample list of research objectives is shown below:

- Gather and process publicly available shale gas data.
- Get insight into the trends of important metrics, specifically the cumulative gas produced after 1-year MCF via exploration of descriptive and advanced statistics.
- Design and train machine learning models to fit the available data.
- Compare two machine learning algorithms (decision tree and random forest) in terms of performance, goodness of fit, explanatory impact, and level of significance.

- Determine, based on numerical simulation scenarios, the total amount of gas produced after a year MCF.

The following is a list of possible research inquiries:

- What are the essential properties of the data, and are there any outliers?
- Are two or more variables related to one another?
- How closely will the predicted model match up with future data?
- What variables, or what combinations of variables, affect shale well performance prediction?

Significance of the Study

This modelling tool can be used as a guide toward a practical gas production decision and also has a procedure for minimizing cost and maximizing productivity. Therefore, it is worth exploring machine learning as a modern technique to identify hidden trends and patterns necessary to resolve this problem.

Limitations

This study is limited only by analyzing independent and dependent variables also a comparative study of machine learning models to identifying trends and patterns in the performance prediction of shale wells by predicting cumulative gas produced after 1-year MCF. While the findings of the study are limited to the data set used, approach taken can be tested with different data sets.

Definition of Terms

- **Data Analytics (DA)** is the research and modeling of undiscovered relationships and patterns in complex, multidimensional data sets utilizing a careful data collection and analysis procedure (Mishra et al., 2021).
- **Machine learning (ML)** is the process by which an equation (commonly referred to as a "black box") is used to deduce the underlying input/output relationship from data (Mishra et al., 2021).
- **Shale Gas:** is a natural gas that is trapped in shale strata that is unconventional.

Summary of the Study

The following guidelines have been stated below to briefly discuss how this project was carried out:

Chapter 1 is an overview of the subject and the primary issue with analyzing and predicting the performance of shale wells.

Chapter 2 contains a review of the literature. This chapter includes a thorough discussion of a literature review that covered a variety of topics relating to the topic, including data analysis and machine learning methods.

Chapter 3 discusses the approach used; in this chapter, all steps necessary to finish the project are covered in depth.

Chapter 4 presents all the results and discussion made from this study. In this chapter, all of the machine learning results are displayed and discussed.

Chapter 5 highlighted the results and recommendations. The concluding thoughts and some suggestions for additional research are presented in this chapter.

CHAPTER II

Literature Review

This chapter briefly describes reservoir modeling and simulation, shale gas reservoirs, horizontal drilling, hydraulic fracturing, Artificial intelligence, data analytics and machine learning.

Reservoir Modeling

In order to estimate reservoir performance under various operating conditions, reservoir simulation is an approach that combines a number of ideas, including mathematics, physics, reservoir engineering, and computer programming (Ertekin et al., 2001). The most important physical processes that happen inside the reservoir system are included in these mathematical equations, including mass transfer between different phases and fluid movement split into the three phases of oil, water, and gas (Ertekin et al., 2001).

Application of Reservoir Simulation in Shale Well Studies

Reservoir simulation models are the main instruments used to carry out the initial research linked to the uncertainty analysis of fluid sequestration. They make it possible to forecast how the storage and injection processes would function under various geological situations (Mohagheh, 2018). The management of natural gas production from unconventional resources like shale is likewise effectively handled by these commercial reservoir simulators (Boosari et al., 2015).

The complexity of the simulation model, however, rises with the length of the run. Any research project needing tens of thousands of simulations, such uncertainty analysis, optimization research, or History matching may become unworkable due to the long run time and high computational effort requirements. The oil and gas industry have long struggled with these protracted execution times of numerical reservoir simulation models (Mohagheh,2018). Complete physics-based simulators are widely used, but this poses a number of problems, including their high processing costs (Schuetter et al,2018). For this reason, in addition to

numerical reservoir simulators, data-driven modeling and comprehension tools for the crucial components of fluid isolation in unconventional reservoirs must be developed and utilized.

Exploratory Data Analysis

EDA is mostly used to get a basic comprehending of the data in the concept of the characteristics of the individual variables and the connections among them. Other objectives include selecting instruments for thorough study, identifying important variables of interest, and developing questions for future data analysis as shown in Figures 2.1 to 2.3. (Mishra & Datta-Gupta, 2018).

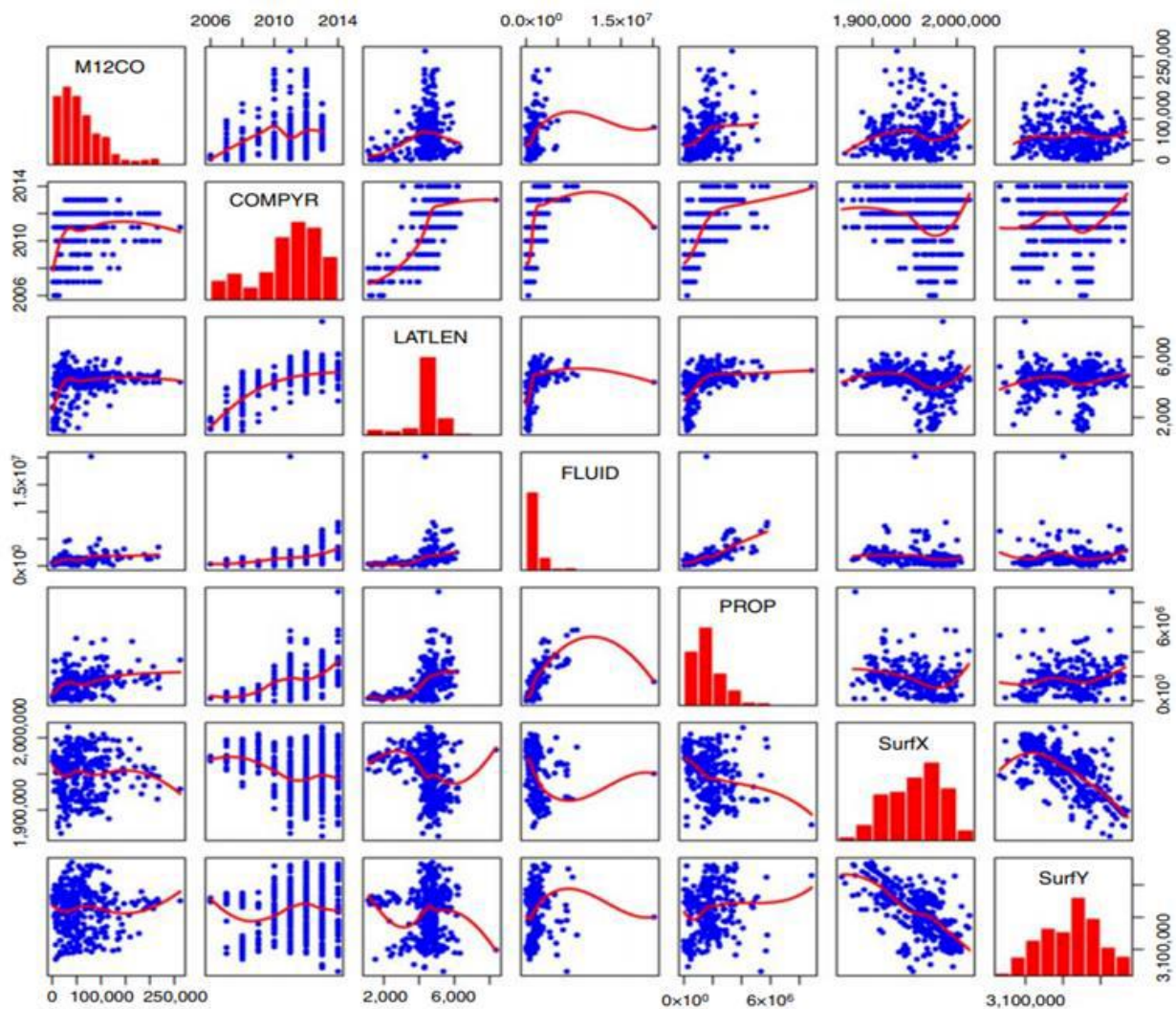


Figure 2.1. Scatter plot matrix (Schuetter et al. 2018)

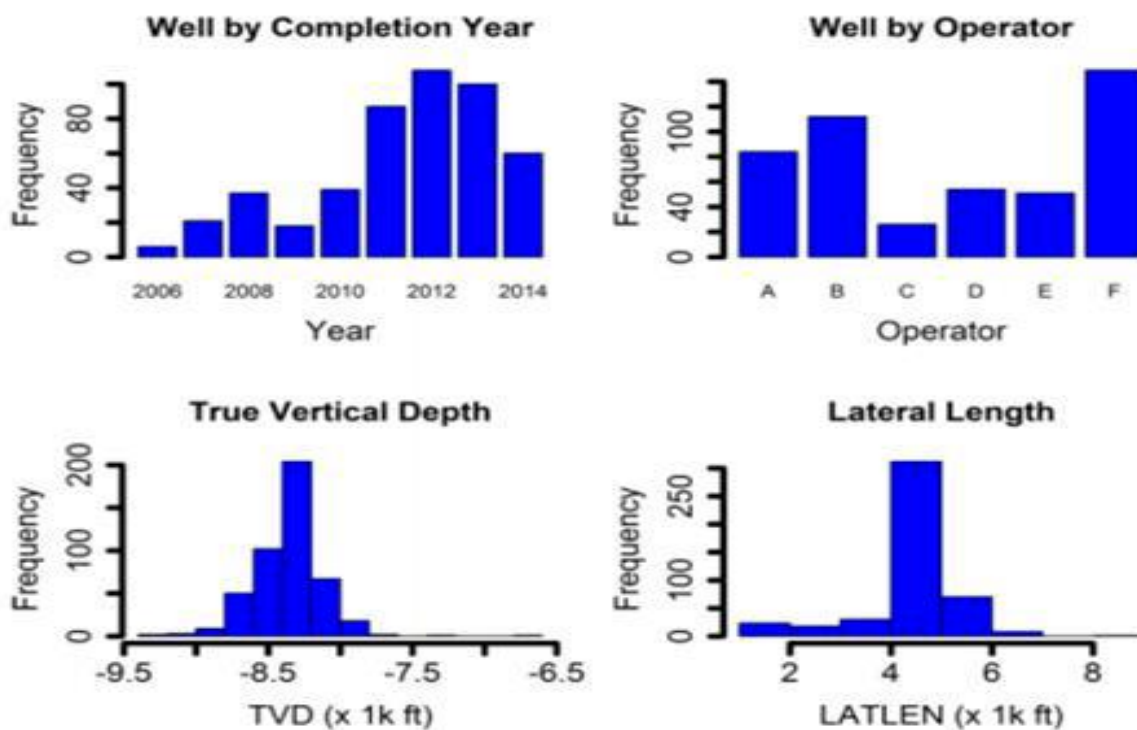


Figure 2.2. Histograms for predictor variables (Zhong et al., 2015)

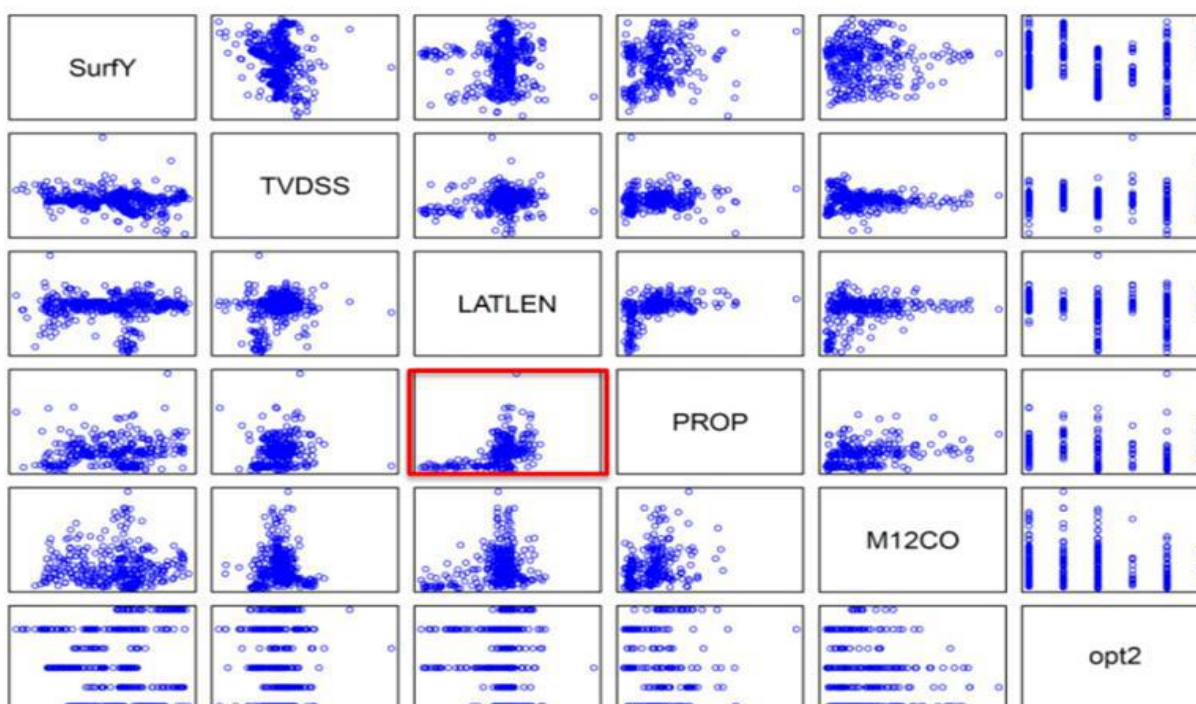


Figure 2.3. Scatterplot matrix for predictor variables (Zhong et al., 2015).

- **Predictive Input and Output Modeling**

Predictive modeling, according to Kuhn & Johnson (2013), is the process of creating a pattern or mathematical tool that makes an accurate forecast. According to Lolon et al. (2016), the model could take the form of a formula or algorithm that has one output variable to forecast and one or more independent, well-known predictors as inputs.

Unconventional Gas Recovery

There is various meaning for an unconventional gas system, typically regarded as unconventional gas reservoirs since it is significantly more difficult to generate from these reservoirs both economically and technically. In addition to these factors, unconventional gas resources are not buoyancy-driven accumulations like conventional gas resources. Most unconventional gas reserves are not affected by stratigraphic or structural traps. The presence of significant amounts of hydrocarbons is another hallmark of unconventional gas reservoir. The development of unconventional reservoirs is challenging, nevertheless. In contrast, despite the smaller size of conventional reservoirs, recovering hydrocarbon is simpler.

Resources for unconventional gas are becoming more crucial for the resource base. For instance, the US uses unconventional gas reservoirs to produce more than 25% of its natural gas. In the ensuing decades, it appears that the production percentages will increase. (Kulga,2010).

Shale Gas Reservoirs

Shale gas is the name given to natural gas that is trapped inside shale rocks. Petroleum and natural gas can be found in large quantities in the fine-grained sedimentary rocks known as shale (see Figure 2.4). The pores in this sedimentary rock are shale gas-filled. Gas in gas shales is frequently kept in three different ways.

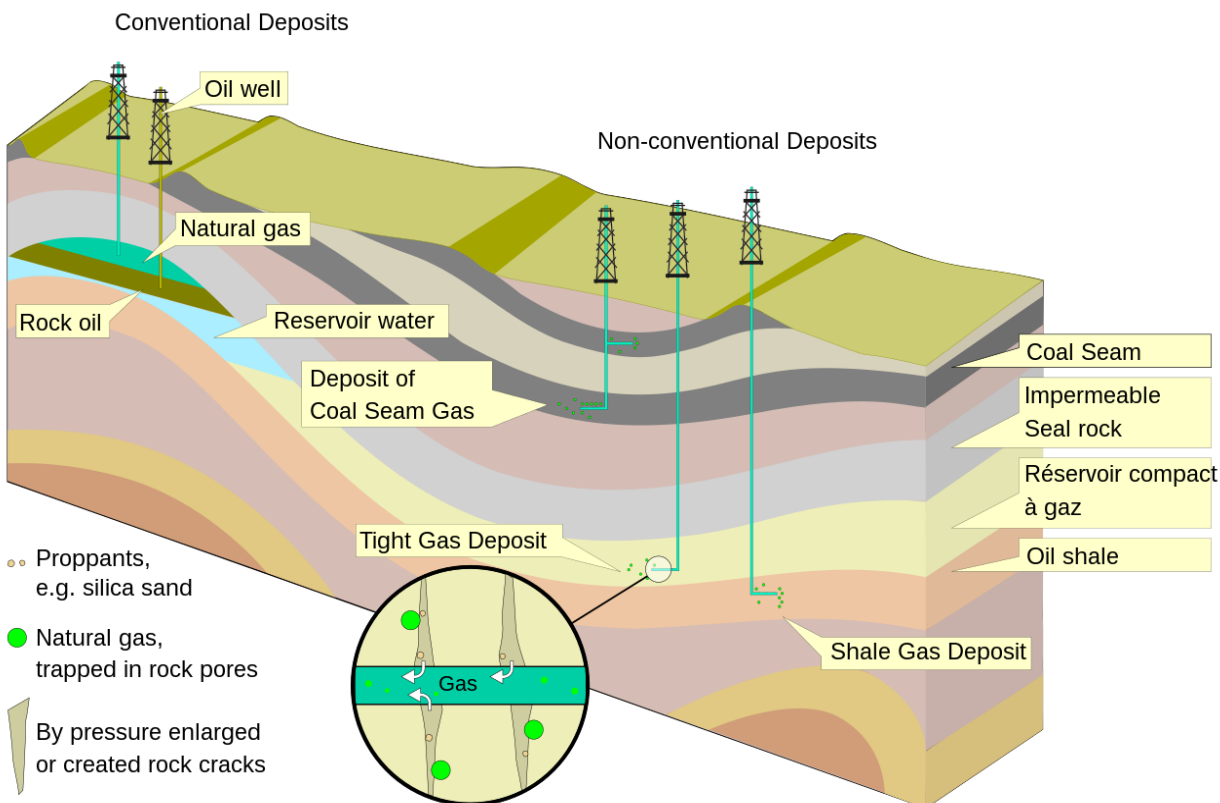


Figure 2.4. An illustration of shale gas compared to other types of gas deposits. (Stevens et Paul, 2012).

In the past ten years, huge amounts of shale gas that were previously uneconomic to generate have been extracted by using the hydraulic fracturing and horizontal drilling. The natural gas business has been rejuvenated by the exploitation of natural gas from shale deposits.

Horizontal Drilling

Horizontal wellbore technology is the first essential component of creating a shale gas reservoir. Since the late 1980s, horizontal drilling technique has been used commercially. Horizontal drilling can be divided into four categories, according to (Lolon,2016). The most common and successful drilling technique has been the medium-radius well. These days, horizontal wellbores can reach lengths of up to 8000 feet. Although drilling and finishing a horizontal well can be more expensive than drilling and finishing a vertical well for production, it can be far more advantageous in other ways. Drilling a vertical well can increase the wellbore

surface area, which is the first principle of drilling a horizontal wellbore. Drilling a horizontal well as opposed to a vertical well has several additional financial advantages.

Hydraulic Fracturing

Stimulating a well using a hydraulic fracturing method is another essential technology for creating a shale gas reserve. In order to increase the effective wellbore radius, hydraulic fracturing is typically done in shale reservoirs with micro-Darcy-range permeability. Fluids and proppant need to be pumped under high pressure in order to fracture the reservoir in order to stimulate a well using the hydraulic fracturing process. Figure 2.5 shows major steps of hydraulic fracturing.

Types of Hydraulic Fracturing

There are two ways that the formation can be hydraulically fractured (Figure 2.5), depending on the orientation of the in-situ stress. Transverse or longitudinal fractures to the horizontal well axis are both possible. A longitudinal fracture is produced if a horizontal well is dug perpendicular to the axis of the least major formation stress. (Zhong, M., Schuetter, J., Mishra, S., & LaFollette, R. F. 2015).

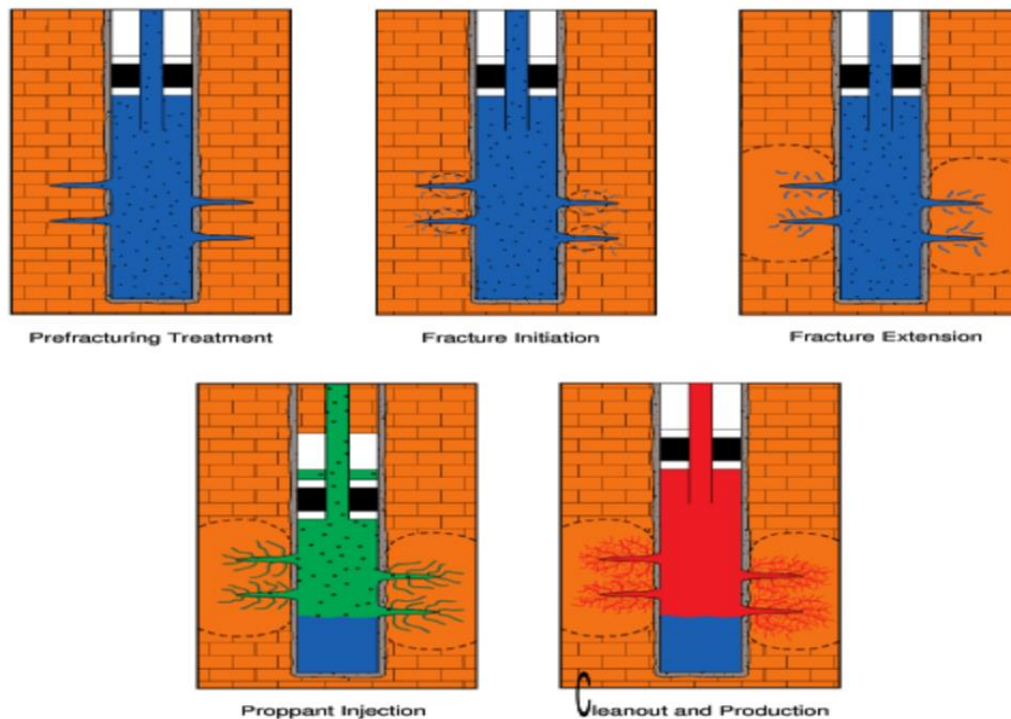


Figure 2.5. Hydraulic fracturing process (introduction to well testing. schlumberger, bath, england, 1998).

Data Mining

Data mining is the require separation of certain information from a database that was previously hidden and not immediately accessible to the user. It uses a variety of methodologies, including machine learning (ML). It is also known as knowledge discovery in databases (KDD). In contrast to data mining, which involves paying someone to find the finest basketball courts, machine learning entails teaching someone how to play the game. Data mining is used by machine learning algorithms to connect many linear and nonlinear connections (Belyadi et al, 2019), data mining techniques can also be used to acquire and aggregate information from websites, online services, and social media.

Artificial Intelligence

Artificial intelligence is essentially the application of machine or computer intelligence as opposed to that of humans or other animals. It is a subfield of computer science that investigates how well computers can mimic cognitive functions.

Machine Learning

Modern data science methodology uses machine learning. To be fair, machine learning has the algorithm at its core, which gives it a major advantage over all other traditional data science methodologies. These are the guidelines a computer follows to locate a model that as closely matches the facts as feasible. In contrast to conventional data science techniques, machine learning uses the algorithm's instructions to teach itself how to locate the desired dependence. This is where machine learning differs from typical data science techniques. Contrary to typical data science, little human interaction is used. In reality, deep learning algorithms in particular are so complex that humans are unable to fully comprehend what is going "inside" the model (365-DS-Booklet).

- **Machine Learning Algorithm**

Each new experiment in a machine learning algorithm is at least as successful as the previous one, which is similar to a trial-and-error process. But keep in mind that a computer needs to make hundreds of thousands of mistakes before it can learn effectively, with the frequency of mistakes decreasing over time. After training, the computer will be able to examine new data and make incredibly precise predictions using the complex computational model it has learnt.

I. Supervised Learning.

Labeled data serve as the cornerstone of supervised learning. Large data can be compared to videos and images with the tags "cats," "dogs," and "other." If the computer's performance doesn't produce the right response, an improvement algorithm changes the computing power, and the computer runs another test. Typically, the computer executes this action simultaneously on a huge number of data points (365-DS Booklet).

II. Unsupervised Learning

When there is not enough time or money to label the data, or when the data scientist is unsure of the precise meaning of the labels, unsupervised learning is utilized. Unlabeled data must be sent to the computer along with guidelines on how to draw conclusions in order to do this. As a result, the data is typically divided into groups according to predetermined criteria. As a result, it is combined. Unsupervised learning is especially effective at identifying data patterns that people using traditional analytical techniques would miss (365-DS-Booklet).

III. Reinforcement Learning

This kind of machine learning prioritizes performance (the capacity to walk, see, and read) over correctness. The computer is rewarded whenever it performs better than it did previously; but, if it performs less than optimally, the algorithms leave the calculation alone. A dog learning commands comes to mind. If the animal follows the rules, it gets a treat; if it doesn't, it doesn't. (365-DS-Booklet).

Programming Languages

Using a programming language, they are comfortable with, the data scientist can write programs that perform certain procedures. The best feature of a programming language is flexibility. R, Python, MATLAB, and SQL are the most often used tools. R and Python are the two most frequently used data science technologies overall. By far, their biggest advantage is their capacity to edit data and integration with different data and data. They are adaptable and not just suitable for calculations involving arithmetic and statistics (365-DS-Booklet).

Software and Frameworks

Excel can be used to manage traditional data, artificial intelligence, and data science. Similar to this, SPSS is a widely used tool for handling conventional data and statistical analysis. On the other hand, Tensor Flow is a software framework and library created specifically for utilizing enormous volumes of data and creating machine learning algorithms. It was created by Google for internal use, became available to the general public in 2015, and now dominates machine learning usage and applications (365-DS-Booklet).

Related Research

Hydraulic fracturing technology has seen a rise in the development of unconventional reservoirs over the past ten years (Muther et al. 2020a, 2020b). Its main applications include the development of tight gas and oil reservoirs as well as the use of horizontal wells for shale gas. Particularly in Canada and North America, this spike was seen. According to Syed et al. (2020a), shale gas made up almost fifty % of all-natural gas produced in the United America State in 2018.

The development of models that can forecast EUR, the generation behavior of unconventional HCs, and give an accurate estimate of the amount of injected fluid or proppant using a variety of supervised learning-based techniques (Kuhn, M., & Johnson, K.,2013). To create AI and ML-based models, researchers with backgrounds in prospection and production as well as AI and ML knowledge joined forces. In the lines that follow, a succinct assessment of the literature is provided, including an estimation of the production performance of shale gas wells using several ML and AI-based methodologies.

ML strategy was the subject of extensive investigation as well. Based on a data-driven methodology, it evaluates the effectiveness of shale gas by considering a number of variables, such as HF and well completion in the Eagle Ford formation. Based on supervised learning, a core machine learning technique, a model prediction of cumulative production data was developed using ML modeling and a data-driven approach. (Han et al. 2020).

General workflow

A data analytics and statistical modeling approach was employed to address the study challenge. Figure 2.6 provides a summary of this approach.

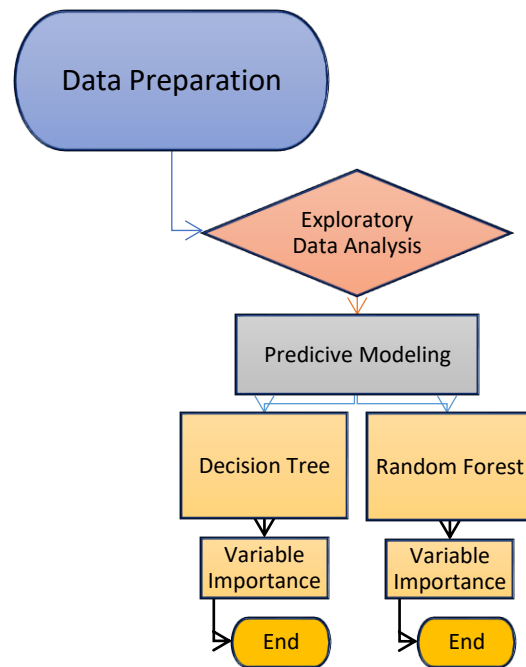


Figure 2.6. Workflow for data analytics approach

Data Preparation

- Perform a quality check on the input data after importing the dataset in Python.

Exploratory Data Analysis

- EDA was used to find hidden patterns and characteristics, like outlier points and the connection between operational and reservoirs parameter.

Predictive Modeling

- Machine learning algorithms were used to predict the total amount of gas generated after 1-year MCF using the reservoir and operational information.
- The models were then evaluated using the goodness of fit approach.

Variable Importance

- Selected the key reservoir and operational characteristics that influence the total amount of gas produced.

CHAPTER III

Research Methodology

This chapter outlines the study population, the research design, the data collection method, the validity and reliability of the data collection instrument, and the data collection process.

Research Design

A structured correlational research design is used to achieve the objective of the study. This design facilitates the description of a situation in its current state, and elicits information directly from the study area.

Population of the Study

The study population consists of a portion of actively producing fields with 49 production wells in total, in four areas with different formation in EAGLE FORD, HAYNESVILLE SHALE, BOSSIER SHALE, MARCELLUS and data was collected from them.

Validity and Reliability Criteria

Machine learning uses a model to train and find relationships between input and output data given to it in a trial-and-error process using an objective function to calculate learning error and optimization algorithm to adjust in order to minimize training errors. Since the algorithm tries to fit a model in a given data, special techniques must be used not to over fit (over train the model) or under fit (not being able to capture the underlying logic) the data as well as the power of prediction. In this regard, validity and reliability criteria are set as follows.

- Field data is pre-processed and made suitable for machine learning.
- In the analysis section a decision tree and random forest will be designed to fit a model into the collected data as intuitively as possible to find trends and patterns.
- Priors were set, the dataset was balanced and splintered into a 70%, 75% and 80% training sets and 30%, 25%, and 20% testing sets.
- After training the model, the 30%, 25%, and 20% testing set will be used to calculate the accuracy of prediction of all data points in the testing set using the trained model.
- 70% accuracy is good and acceptable for further predictive analytics, 90% and above is impressively okay.

Method of Data Collection

- Data collected would be used to create a database table in excel.
- Data redundancy would be normalized and balanced.
- A file extension with xlsx will be used to facilitate IO operations in data science frameworks.

Data Analysis Technique (Exploratory Data Analysis of Data Set)

- Raw data from excel file format (suitable file type for jupyter notebook)
 - a. Explore descriptive statistics.
 1. Mean, standard deviation, mode.
 2. Quartiles (Q1, Q2, Q3).
 3. Minimum and maximum values.
 4. Counts.
 - b. Conduct advanced statistical tests.
 1. Significance test (p-values).
 2. Calculating explanatory power of correlated variables (R-squared & adjusted R-squared).
 3. Measure of overall significance (F-statistic and F-probability).
 4. Skewness and kurtosis.
 - c. Assumptions with linear machine learning regression
 1. Linearity
 2. Homoscedasticity/ Heteroscedasticity
 3. No autocorrelation
 4. Normality
 5. Multicollinearity

Machine Learning (Supervised Learning Algorithm)

Data, models, objective functions, and optimization algorithms must all be specified when creating a machine learning algorithm. A model gives the machine learning algorithm a sense of direction to train and learn on its own. An objective function estimates error after each trail of the training process. An optimization algorithm is used to find the objective function's minima in order to minimize error and improve accuracy.

Metrics for Evaluating Regression

Before going any farther with the implementation of a Decision Tree and Random Forest model in scikit-learn, let's go through some of the regression assessment metrics outside of R and R^2 . (Scikit-learn, 2020)

1. **Mean absolute error (MAE):** is the average error value in absolute terms. It is merely the average of the absolute difference between the numbers that were forecasted and those that actually occurred. Since the objective is to minimize this loss function, MAE is also known as a loss function and is defined as follows:

$$\mathbf{MAE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i) \quad \mathbf{3.1}$$

Where

y_i True response

\hat{y}_i Predicted response

2. **Mean squared error (MSE):** It is known as the mean of the squared error, as suggested by the name, as can be seen in the example below. There is another loss function called MSE that also needs to be minimized. Due to the fact that MSE's objective function penalizes greater errors more harshly than MAE does, MSE is frequently utilized in real-world ML applications (Scikit-learn, 2020).

$$\mathbf{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad \mathbf{3.2}$$

Where

y_i True variable

\hat{y}_i Predicted variable

3. **Root mean squared error (RMSE):** As may be seen in the graphic below, RMSE is essentially the square root of MSE. Please be aware that due to its interpretability, RMSE is another very well-liked loss function. (Scikit-learn, 2020).

$$\mathbf{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad \mathbf{3.3}$$

Where

y_i True variable

\hat{y}_i Predicted variable

Tree Methods

- **Decision Tree**

A supervised ML approach known as a decision tree can be applied to classification and regression issues. It is essential to comprehend how a decision tree functions before talking about decision trees and random forest which are related concepts. The data is separated into sub trees using a decision tree, which are further subdivided into sub trees. As shown in Figure 3.1, while a terminal node, also known as a leaf node, is the lowest node and no longer splits, a decision node, also known as an internal node, comprises two or more branches. The root node, which is the highest level, accurately represents the entire population. You should be aware that "splitting" is the splitting of a node into two or more sub nodes.

There are various decision tree algorithms. The 1986 Iterative Dichotomize 3 algorithm, generally known as ID3 (Quinlan, R.1986), was developed. This approach, which will be explained, builds decision trees utilizing categorical traits in a top-down greedy manner to maximize information gain. An ID3 multi-way tree is also used. It is not necessary for features to be categorical when using C4.5, another decision tree approach. In comparison to C4.5, the most recent Quinlan release, C5.0, uses less RAM and produces smaller rule sets.

Finally, C4.5 is similar to CART but includes numerical target variables and does not compute rule sets. CART define classification and regression trees. The highest information gain at each node is achieved by CART's creation of binary trees. Please be aware that the CART algorithm is used in an optimal manner by the scikit-learn library (Scikit-learn,2020).

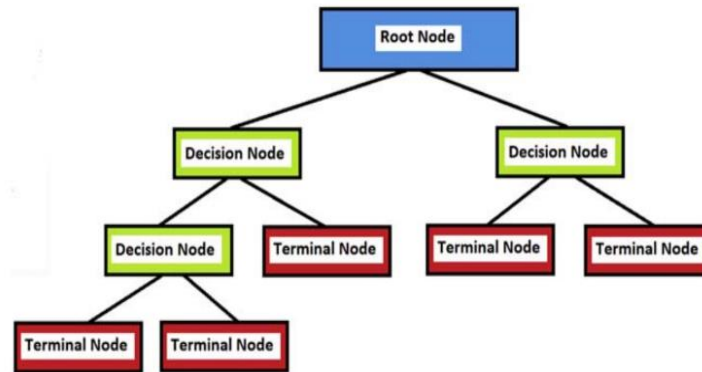


Figure 3.1. Decision tree illustration. (Breiman, L. 2011).

- **Attribute Selection Technique**

Determining which attributes belong at the root or internal node of a data collection with N attributes can be difficult and complex. The following list includes some of the most crucial factors for attribute selection:

1. **Entropy:** Entropy is only a calculation that measures uncertainty or purity. Keep in mind that low purity means high entropy.

$$E(S) = \sum_{i=1}^C -P_i \log_2 P_i \quad 3.4$$

Where

P_i is the chance that a class will appear in a dataset,

C is the total number of classes.

Entropy for many qualities can be determined mathematically as follows:

$$E(X, Y) = \sum_{c \in Y} P(c)E(c) \quad 3.5$$

Where

X is the present situation.

Y Is the chosen attribute.

P(c) is the attribute's probability.

E(c) stands for the attribute's entropy.

2. **Information Gain (IG):** When creating a decision tree, it's critical to identify the attribute that yields the highest information gain and the lowest entropy. A characteristic's ability to successfully place training cases in the appropriate category is described by IG. IG favors smaller partitions, which can be determined by the following:

$$\mathbf{IG (Y, X) = E(Y) - E (Y, X)} \quad \mathbf{3.6}$$

3. **Gini Index:** As demonstrated here, the Gini index is determined by deducting 1 from the sum of squared probabilities for each class. The Gini index encourages bigger partitions as opposed to information gain. Please note that the Gini index would be zero in fully categorized samples.

$$\mathbf{Gini = 1 - \sum_{i=1}^c (Pi)^2 = 1 - (P(class A)^2 + (P(class B)^2 + \dots + P(class N)^2)} \quad \mathbf{3.7}$$

Where

Pi is the probability of an element being classified under a particular class.

Over fitting is one of the most challenging problems when employing a decision tree. Pruning is one strategy for preventing over fitting. Pruning is the act of removing branches or tree trunk segments that don't offer much information for classifying cases or that otherwise interfere with overall accuracy. To ensure that the model is not over fitted when employing a decision tree, cross validation is also crucial. A different strategy is to employ the algorithm of random forest, which typically performs better than a decision tree. This chapter's next section goes into great detail on

random forests. When selecting attributes, the previously mentioned criteria, including knowledge gain and the Gini index, are utilized to determine the values of each property. Assume that the selection criterion for attributes is information gain. After sorting the data, the attributes with the greatest values are positioned at the root.

- **Random Forest**

A robust supervised machine learning method called random forest as shown in Figure 3.2, can be applied to both classification and regression problems. Ho introduced the widespread application of random decision trees in 1995 (Kam Ho, 1995). A forest of decision trees, or random forest, is a collection of decision trees. Random forest is an ensemble method since it combines multiple decision tree models into a single model. For instance, because projections can differ, MCF might utilize a single tree instead of a decision tree and end up with an inaccurate estimate of the Cumulative Gas Produced after 1 Year. One method to reduce this volatility in estimating the Cumulative Gas Produced after 1 year, MCF, is to use predictions from hundreds or thousands of decision trees and calculate the final result using the average of those trees. The fundamental idea behind random forest is to build a single model out of many decision trees. (Breiman, L.2011)

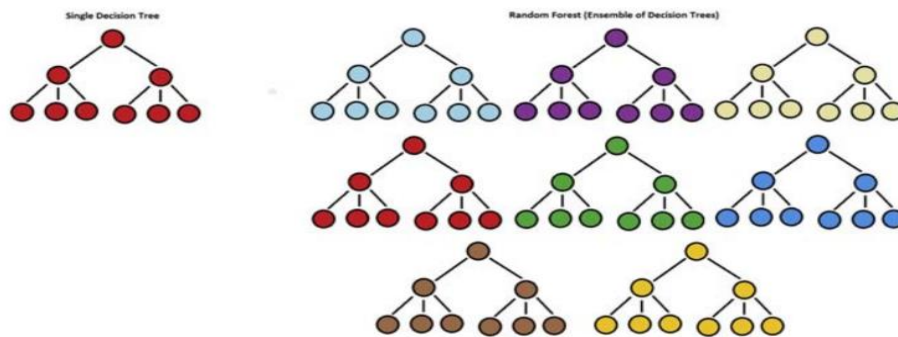


Figure 3.2. Decision tree versus random forest.

While individual decision tree forecasts may not be precise, aggregated forecasts have a higher likelihood of being so. Since it combines substantially more data from several forecasts, random forest typically outperforms a single decision tree in terms of accuracy. Random forest makes its final prediction in regression issues by averaging the decision trees. As was previously mentioned,

classification issues can also be resolved utilizing random forests by polling the vast majority of the anticipated class.

Implementation of Decision Tree and Random Forest Using Scikit-learn

The output feature in this section, cumulative Gas Produced after 1 Year, is linked to a database with the input features stated below. To predict Cumulative Gas Produced after 1 Year, MCF, a supervised regression decision tree and random forest model are being built.

- The input features are initial pressure estimate(psi), reservoir temperature(F), net pay (ft), porosity, water saturation, oil saturation, gas saturation, gas specific gravity, CO₂, N₂, TVD (ft), spacing, stages, number of clusters, clusters per stage, Total proppant (Lbs.), lateral Length(ft), Top Perf(ft), bottom perf(ft), sand surface temperature, (deg F), static wellhead temp (deg F)
- The output feature is Cumulative Gas Produced after 1 year, MCF.

Below is an outline of applicable methods (the training and testing set must be done separately for DT and RF in python jupyter notebook).

- Raw data Pre-processing
 1. Import libraries (NumPy, matplotlib, pandas, seaborn, and from sklearn. Model selection import train_test_split).
 2. Define the x and y variables
 3. Next, import Decision Tree Regressor and Random Forest Regressor from sklearn.
 4. Define the decision tree and Random Forest
 5. Apply dtree and rf to “(X_train. Y_train)”
 6. Obtaining the training and testing R^2 is the next step
 7. Optimized. Next, compare training actual results to those from predictions and testing.
 8. Let's additionally include MAE, MSE, and RMSE to adequately assess the model from all angles.

CHAPTER IV

Results and Discussion

The main goal of this chapter is to explain the results from the analysis and performance predictions of shale wells using data analytics and machine learning techniques. The following cases were examined for this analysis:

- Using descriptive statistics to comprehend and interpret the data
- Use box plots, histogram plots, scatter plots, and multivariate correlation graphs to provide a visual analysis.
- Use supervised learning techniques to predict the cumulative gas produced after a year, such as decision trees and random forests.

Descriptive Statistics

When analyzing a dataset, you should first get a sense of it by posing questions like the following (Hold away, 2009):

- What are the smallest and greatest values?
- What single representative number would be adequate for this batch of data?
- How broad is the spread or variance?
- Does the dataset have a uniform distribution throughout a range of values, or are certain values grouped around others?

Because they describe the data, descriptive statistics and summary statistics can provide answers to these problems. In this study, both the reservoir and the operational parameters were subjected to descriptive statistics.

Reservoir and Operational Parameters

The count, mean, standard, minimum, 25%, 50%, 75%, and maximum values that were obtained from this investigation are shown in Table 4.1 along with the summary statistics for the operational and reservoir parameters.

The first thing to notice is that Table 4.1's mean porosity values are lower (7%), which indicates that the shale rock has been compressed as a result of the stress and has less pore space as a result.

The fact that the average water saturation is lower (0.301) indicates that the insoluble nonconductive substance kerogen is present, which causes a decrease in rock conductivity and an underestimating of water saturation. Additionally, oil saturation is lower (0.10) and temperature-dependent. Any given oil will have a lower saturation point as temperature rises. In our data, we have a value of 0.10 and a mean value of CO₂ (0.01) and N₂ (0.00), and the critical gas saturation ranges between 0.5 and 50% depending on parameters like rock and fluid qualities. Although these values are insignificant, they can be employed to prevent the pore pressure in shale from decreasing and to keep it pliable.

The high mean value of the initial pressure estimate (6313.78 psi), which is also visible in Table 4.1, indicates the amount of driving power that may be used to force the remaining fluid out of the reservoir during a production sequence.

Finally, we have a high mean value of TVD, lateral length, bottom perforation which define the horizontal well operation in shale well.

Different kinds of tales can be told between characteristics and the goal variables using plotting and data visualization. The oldest and most significant area of data science is plotting, or data visualization. And to see how each variable in this thesis is distributed, we will utilize boxplot, histogram plot, scatter plot, and multivariate correlation plot.

Table 4.1 Descriptive statistics for reservoir and operational parameters

	count	mean	std	min	25%	50%	75%	max
Initial Pressure Estimate (Psi)	50	6313.78	2963.47	2200.00	4300.00	5164.00	9929.25	12223
Reservoir Temperature (degF)	50	211.16	93.43	115.00	134.00	144.50	323.00	379.00
Net Pay (ft)	50	163.38	56.79	56.00	136.00	164.50	208.75	268.00
Porosity	50	0.07	0.01	0.05	0.06	0.07	0.08	0.10
Water Saturation	50	0.30	0.08	0.18	0.21	0.31	0.36	0.47
Oil Saturation	50	0.10	0.24	0.00	0.00	0.00	0.00	0.74
Gas Saturation	50	0.59	0.27	0.00	0.57	0.67	0.79	0.81
Gas Specific Gravity	50	0.61	0.09	0.57	0.57	0.57	0.59	0.95
CO2	50	0.01	0.01	0.00	0.00	0.00	0.02	0.05
N2	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TVD (ft)	50	9020.33	2160.28	5707.63	7389.31	7794.69	11755.37	12668
Spacing	50	1220.20	297.46	700.00	1000.00	1200.00	1500.00	1850.00
Number Of stages	50	45.18	20.07	7.00	30.25	47.00	62.50	89.00
Number of Clusters	50.	276.46	160.97	49.00	141.00	234.00	444.00	735.00
Number of Cluster per Stage	50	6.07	2.16	3.00	5.00	5.00	7.00	15.00
Number of Total Proppant (MM Lbs)	50	20.53	8.95	3.59	14.11	20.64	26.53	42.94
Lateral Length(ft)	50	7867.84	2354.97	2268.00	5990.00	7480.00	9800.00	13011.00
Top Perf(ft)	50	9204.48	2224.97	5900.00	7548.50	8133.00	12082.00	13153.00
Bottom Perf(ft)	50	17054.92	3608.28	10049	14360.75	16192.00	20089.50	23203.00
Sand face Temp (deg F)	50	209.61	91.38	115.00	133.81	143.18	303.75	379.00
Static Wellhead Temp (deg F)	50	95.91	49.34	60.00	65.00	80.00	120.00	236.00
Cumulative Gas Produced after 1 year, MCF	50	4378.22	3273.30	25.12	1618.66	3792.71	6355.90	13094.84

Univariate Data Analysis

The basic methods employed for univariate data analysis include scatter plots, histograms, and box plots. We may assess the symmetry of the data and the degree of skewness, as well as whether it contains any outliers, by the visual inspection of these methods.

Boxplots for Reservoir and Operational Parameters

We can get a general notion of the data distribution using a box plot. The data is more compact if the box plot is not too long. The data is dispersed if the box plot is relatively tall. Each of the box plot of 22 variables can be interpreted in terms of the data's spread or compactness. We might use the chart to determine the potential existence of outliers using a broad definition of outliers. Outliers are typically anticipated when there are several data points.

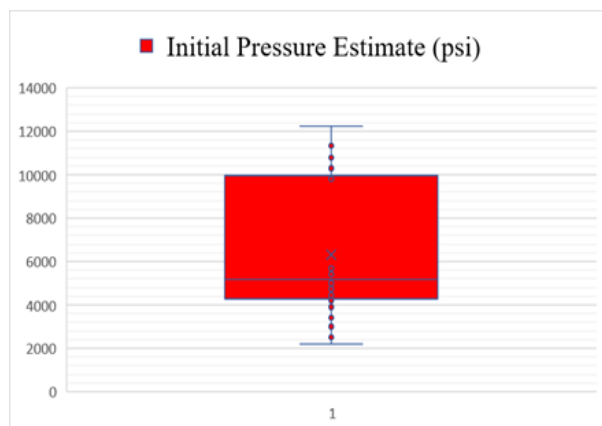
Figure 4.2 shows that the Total Proppant (MM Lbs.) box plot's interquartile range (IQR) falls in the middle of the median for the reservoir parameter box plots. Inferring further that the sample values for the reservoir parameters are distributed equally on both sides of the median, this shows that the sample values are evenly distributed between the median and the IQR.

However, some of the variables plotted in Figure 4.1 reveal the median is located farther from the upper half of the box plot (third quartile). Given that the upper whisker is longer than the lower one, the upper tail of the data is likely to be longer than the lower tail. The box plot is being pulled higher by the variable values. The variability of those variables is increased as a result.

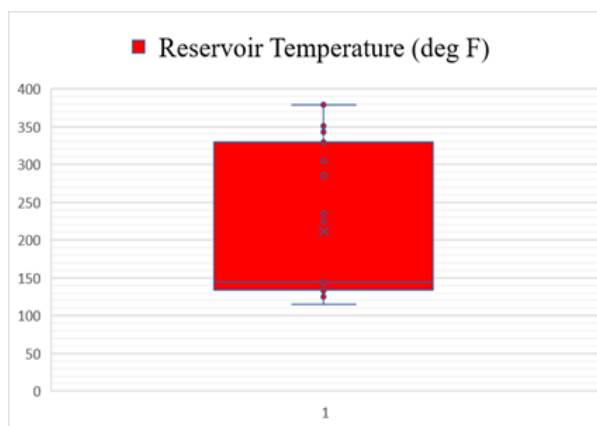
Furthermore, since the top whisker is shorter than the bottom one, we may say that there is less fluctuation of variable. You might use the chart to determine the potential existence of outliers using a broad definition of outliers. Outliers are typically expected with big data points, and we can see their existence in our reservoir and operating parameters box plots (Figures 4.1 and 4.2). Such is the cluster per stage boxplot, gas specific gravity, N₂, and oil saturation. The box plot of Gas specific gravity and N₂ also reveals outliers at the upper end of the data range. If the mean value is above the median, the median line does not divide the box evenly, and the upper tail of the boxplot is longer than the lower tail, the population distribution from which the data were sampled may be skewed to the right.

Boxplots for Reservoir Parameters

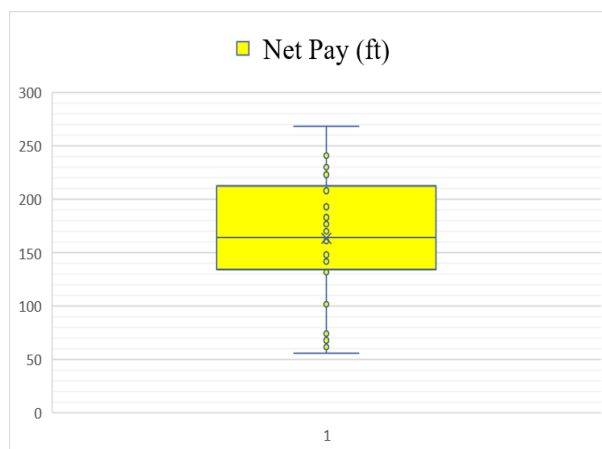
(a) Initial Pressure Estimate (psi) Boxplot



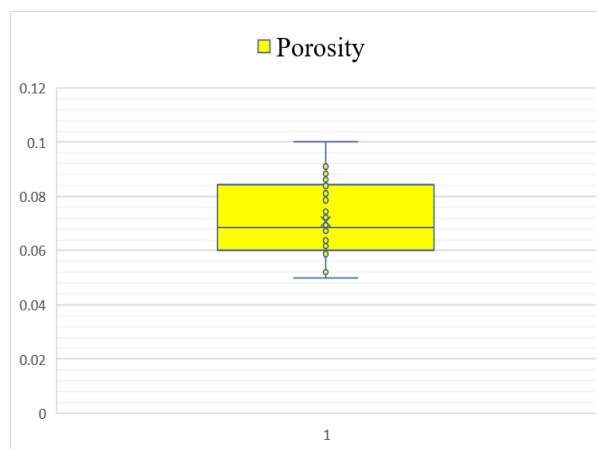
(b) Reservoir Temperature (F) Boxplot



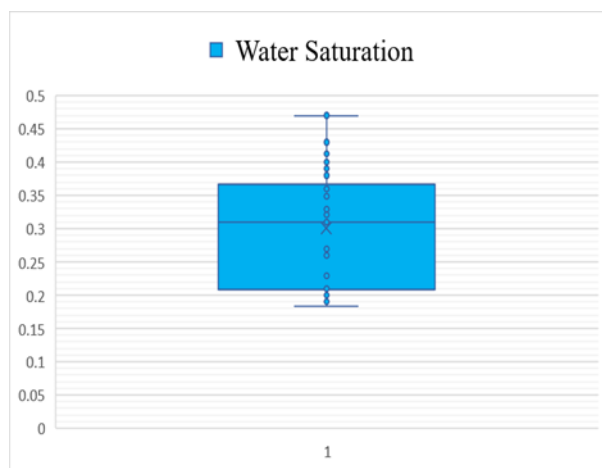
(c) Net Pay (ft) Boxplot



(d) Porosity Boxplot



(e) Water Saturation Boxplot



(f) Oil saturation Boxplot

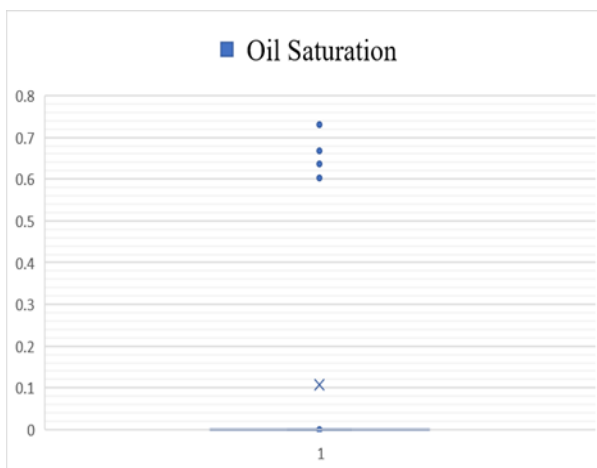
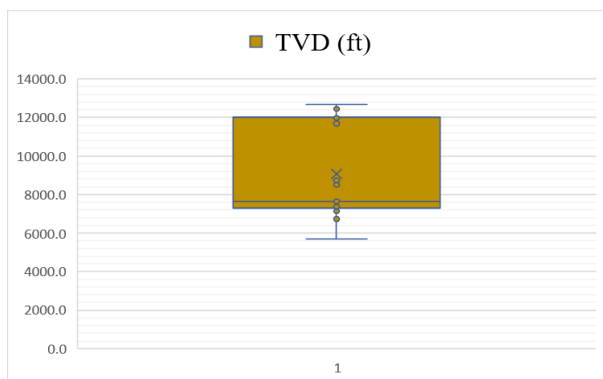




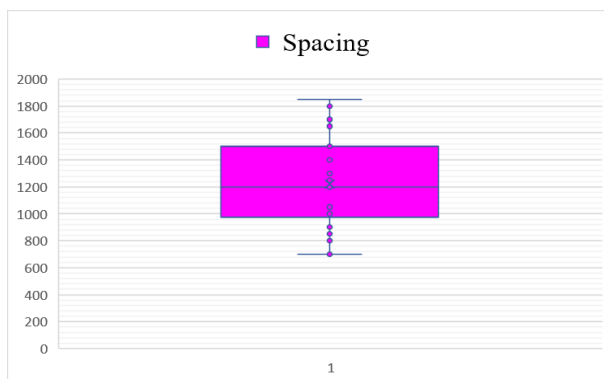
Figure 4.1. Reservoir parameters box plots: (a) Initial Pressure Estimate (psi), (b) Reservoir Temperature(F), (c) Net Pay(ft), (d) Porosity, (e)Water Saturation, (f) Oil saturation, (g) Gas Saturation, (h) Gas Specific Gravity, (j) CO2 (k) N2

Boxplots for Operational Parameters

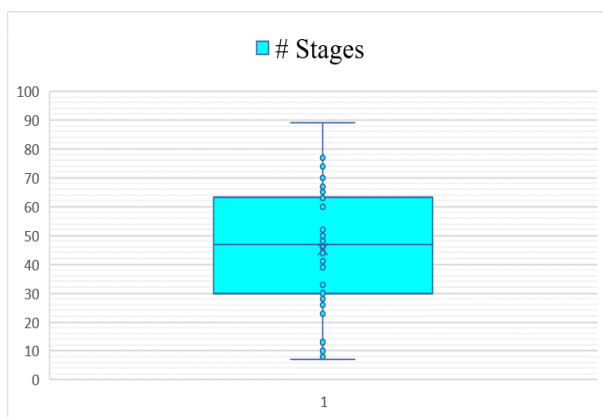
(a) TVD Boxplot



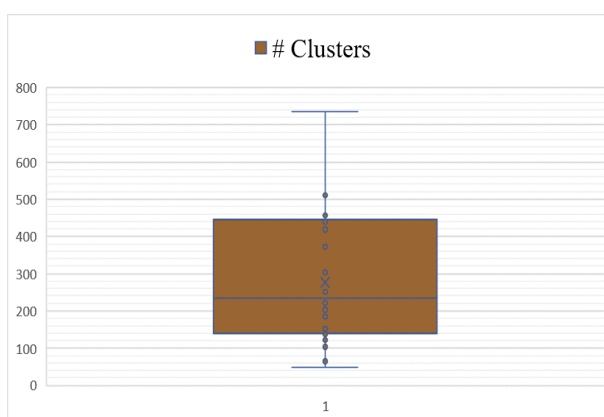
(b) Spacing Boxplot



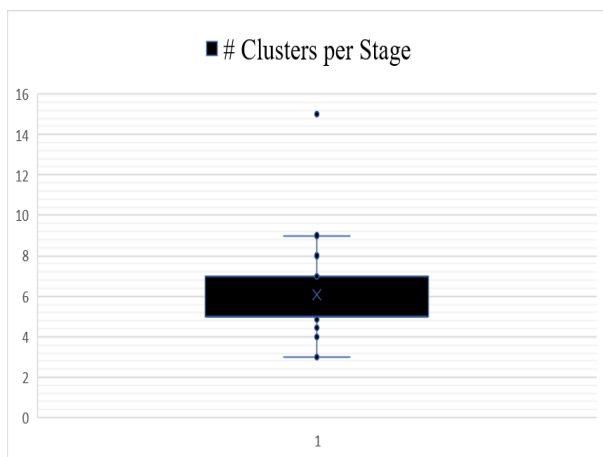
(c) Nummer of Stages Boxplot



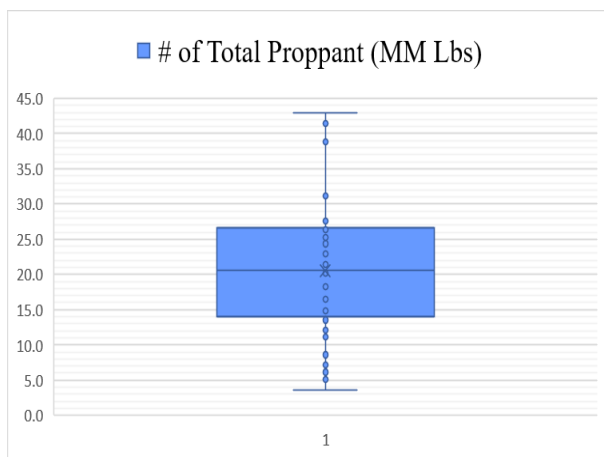
(d) Number of clusters Boxplot



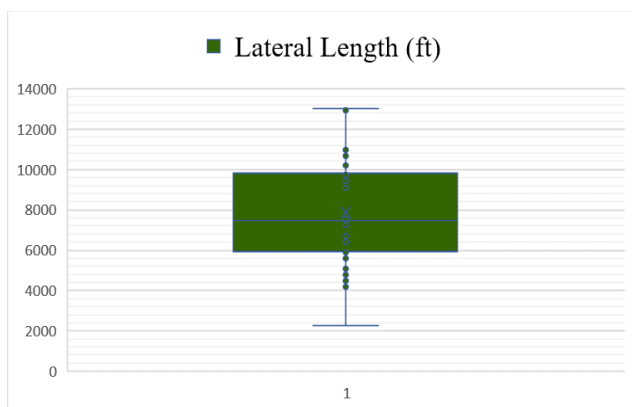
(e) Number of clusters per Stage Boxplot



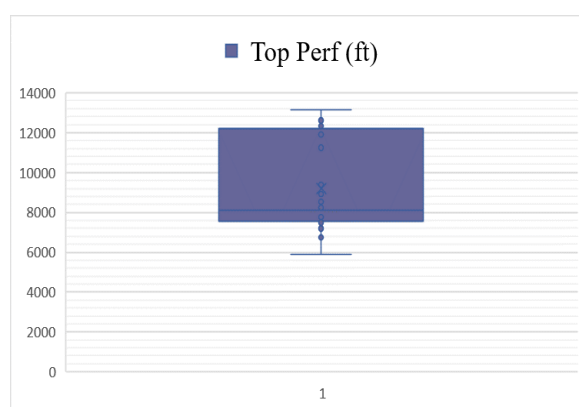
(f) Total Proppant (MM Lbs.) Boxplot



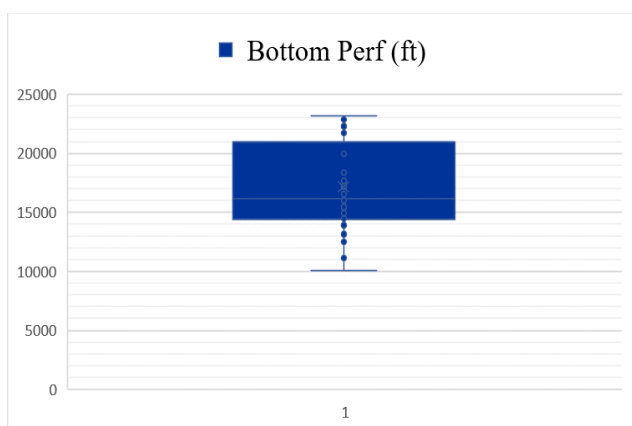
(g) Lateral Length (ft.) Boxplot



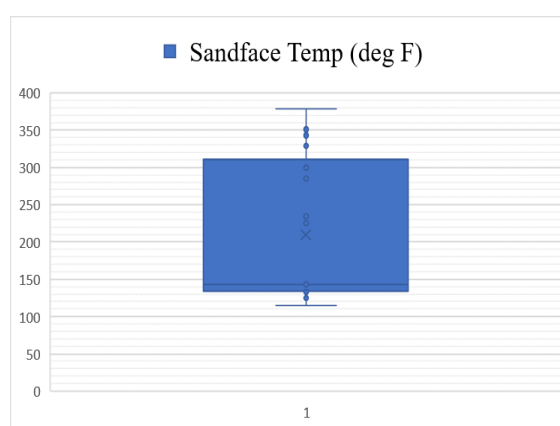
(h) Top Perf (ft.) Boxplot



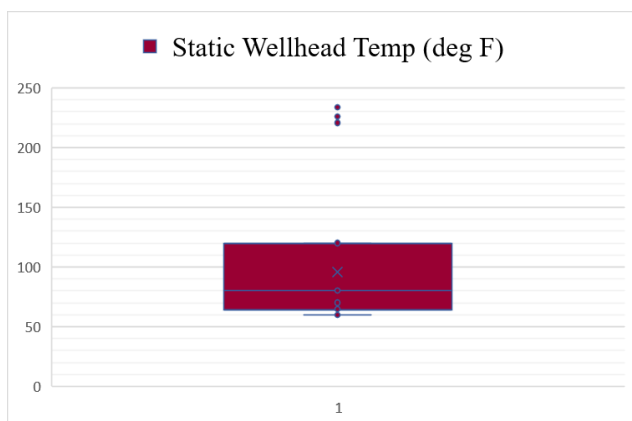
(i) Bottom Perf (ft.) Boxplot



(j) Sand face Temp (F) Boxplot



(k) Static wellhead Temp (F) Boxplot



(l) Cumulative Gas Produced after 1 year, MCF Boxplot

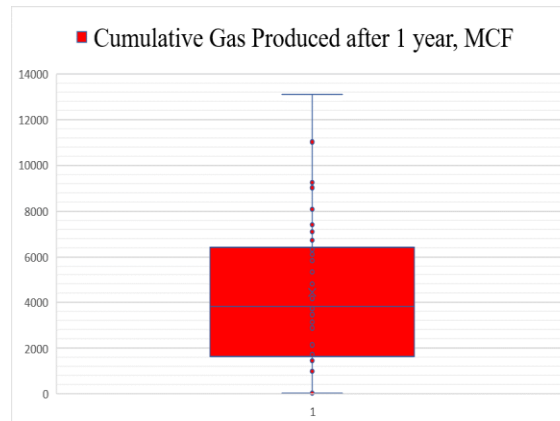


Figure 4.2 Operational parameters box plots: (a) TVD

(b) Spacing, (c) Number of Stages, (d) Number of clusters, (e) Number of clusters per Stage, (f) Total Proppant (MM Lbs.), (g) Lateral Length (ft.), (h) Top Perf (ft.), (i) Bottom Perf (ft.), (j) Sand face Temp (F), (k) Static wellhead Temp (F), (l) Cumulative Gas Produced after 1 year, MCF

Histograms for Reservoir and Operational Parameters

By comparing the lengths of the tails, a histogram can reveal whether the data is skewed. If the right tail is longer than the left tail, the data is skewed to the right; conversely, if the left tail is longer than the right, the data is skewed to the left.

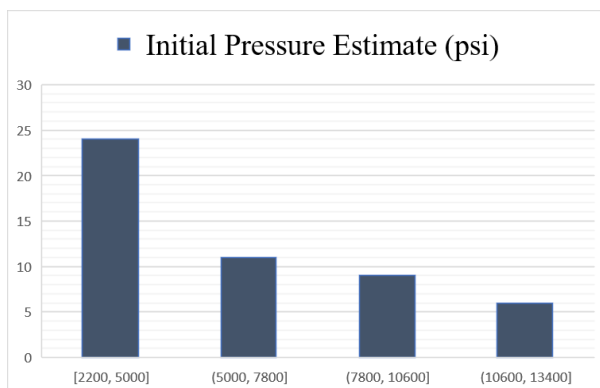
It is clear from Figures 4.3 and 4.4 that the majority of the operational parameters are not symmetrical in terms of the histogram's shape. They all do, to some extent, show skewness. These histograms clearly depict that most of the sample values are at the left and the right side of the tail is longer. Figures 4.3 and 4.4 show that the majority of histograms are not symmetric. A histogram is considered to be positively skewed (skewed to the right) if the tail on the right is lengthy. The median value is lower than the mean, as shown by this histogram.

A lower boundary in a data set is typically the cause of right-skewed data, whereas a higher barrier causes left-skewed data. Therefore, the data will skew right if the lower bounds of the data set are extremely low in comparison to the remainder of the data.

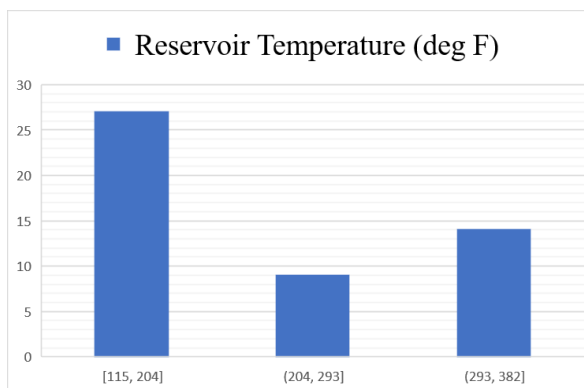
Last but not least, the left-skewed distribution in Figure 4.4 Lateral Length (ft.) Histogram is longer on the left side of its peak than on its right. Negative skew is another name for left skew. That indicates that a higher border is to blame.

Histograms for Reservoir Parameters

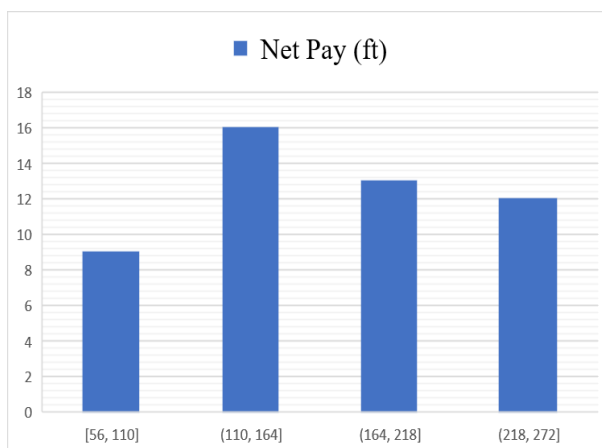
(a) Initial Pressure Estimate (psi) Histogram Plot



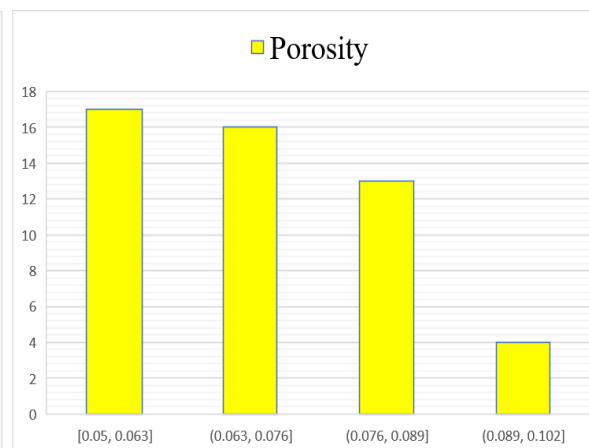
(b) Reservoir Temperature (F) Histogram Plot



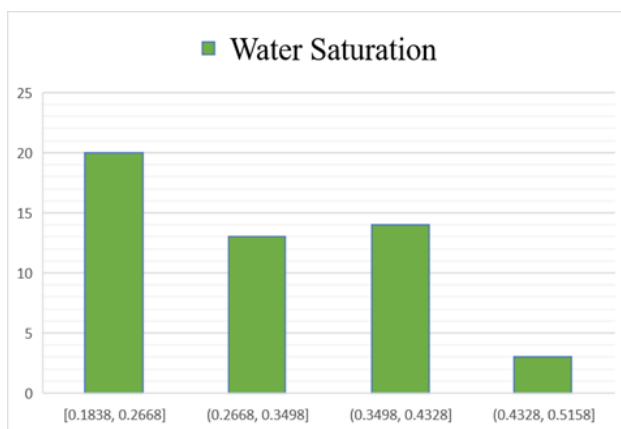
(c) Net Pay (ft) Histogram Plot



(d) Porosity Histogram Plot



(e) Water Saturation Histogram Plot



(f) Oil Saturation Histogram Plot

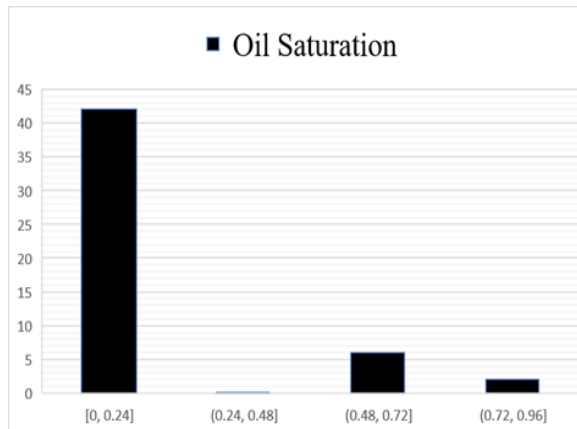
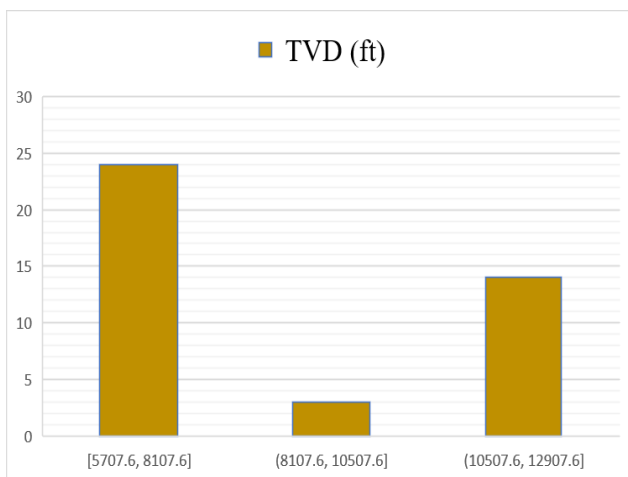




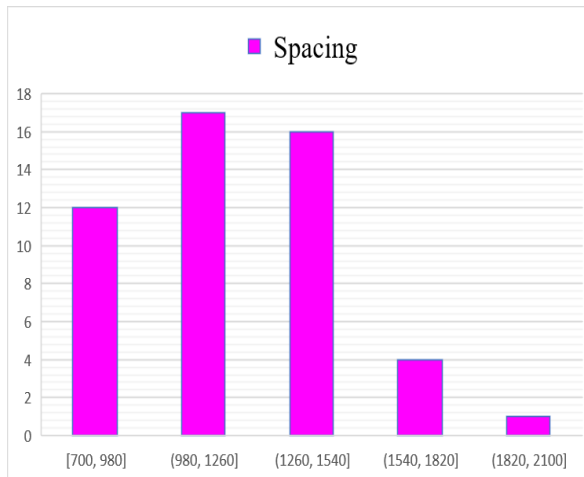
Figure 4.3. Reservoir parameters histogram plots: (a) Initial Pressure Estimate (psi), (b) Reservoir Temperature (F), (c) Net Pay (ft), (d) Porosity, (e) Water Saturation, (f) Oil saturation, (g) Gas Saturation, (h) Gas Specific Gravity, (i) CO₂, (j) N₂

Histograms for Operational Parameters

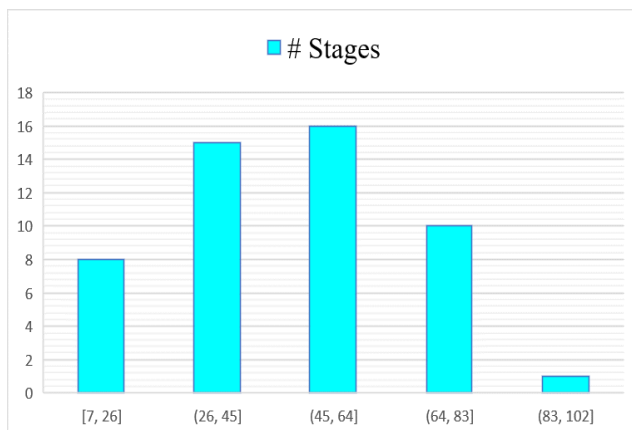
(a) TVD (ft) Histogram Plot



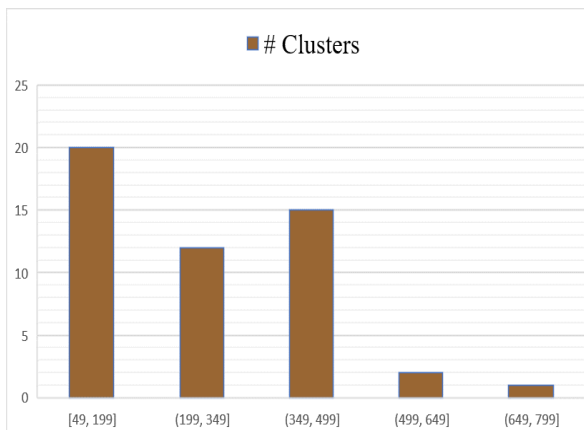
(b) Spacing Histogram Plot



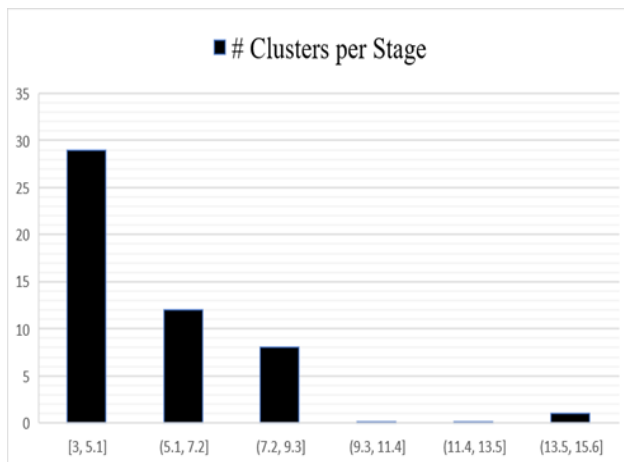
(c) Number of Stages Histogram Plot



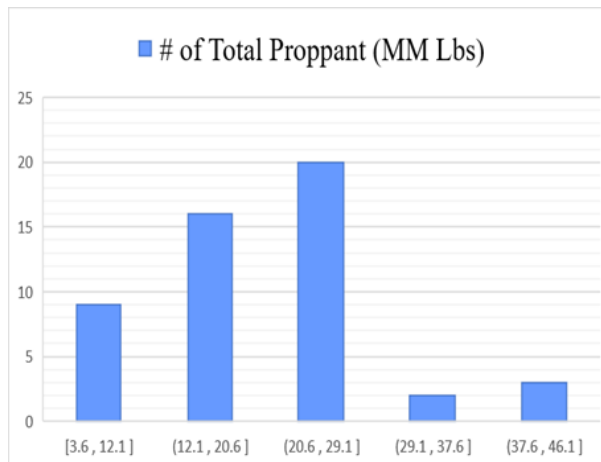
(d) Number of clusters Histogram Plot



(e) Number of clusters per Stage Histogram Plot



(f) Number of Total Proppant (MM Lbs.) Histogram Plot



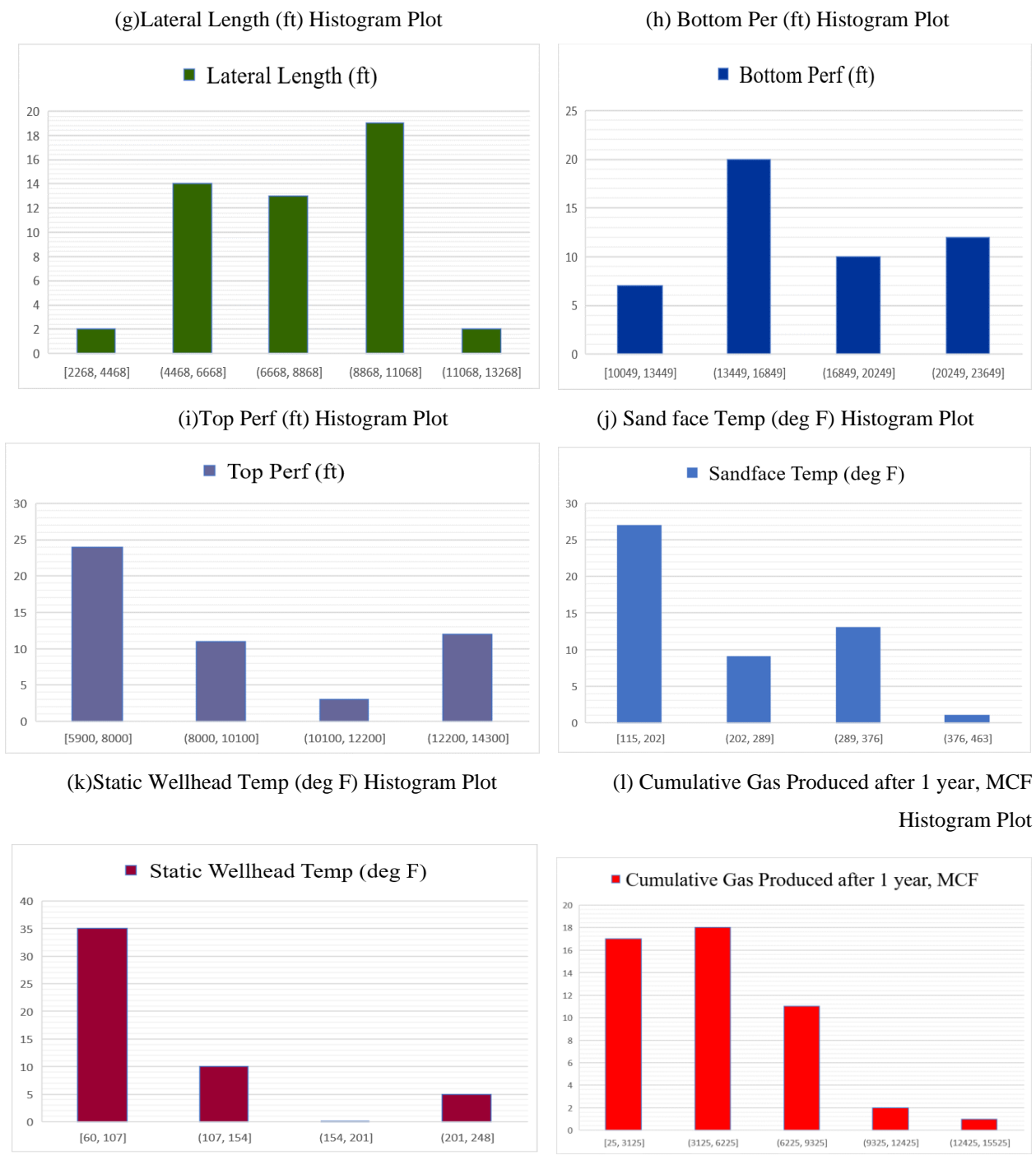


Figure 4.4. Operational parameters histogram plots: (a) TVD
 (b) Spacing, (c) Number of Stages, (d) Number of clusters, (e) Number of clusters per Stage, (f) Total Proppant (MM Lbs.), (g) Lateral Length (ft.), (h) Top Perf (ft.), (i) Bottom Perf (ft.), (j) Sand face Temp (F), (k) Static wellhead Temp (F), (l) Cumulative Gas Produced after 1 year, MCF

Scatterplot for Reservoir and Operational Parameters

In a scatter plot, values for one or more different numerical variables are represented by dots. Each dot's location on the horizontal and vertical axes represents a data point's values. To view relationships between variables, utilize scatter plots.

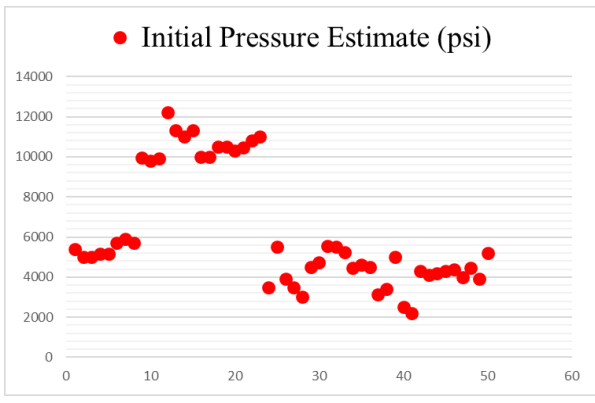
The data in the scatterplots in Figures 4.5 and 4.6 with no association between the variables shows neither positive nor negative trends. The scatterplot displays random, non-directional points. In addition, a scatterplot with no linear trend (positive or negative) is referred to as having a zero correlation or a near-zero correlation.

Also, we can see that none of the data have been modeled, so the scatter plot can only be fit approximately by a linear function because the straight line will pass through all points. We cannot use a scatter plot that shows no association to make a prediction. Because association describes how sets of data are related and when there is no association that means that there is no relationship between them.

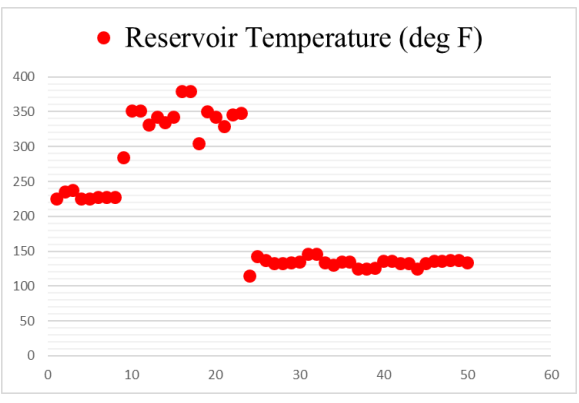
Finally, as the scatter plots have been plotted individually it is complicate to see the association and correlation of data and to make it easy, we are using the multivariable correlation plot to see the high correlation between each variable (Figures 4.7 and 4.8)

Scatterplot for Reservoir Parameters

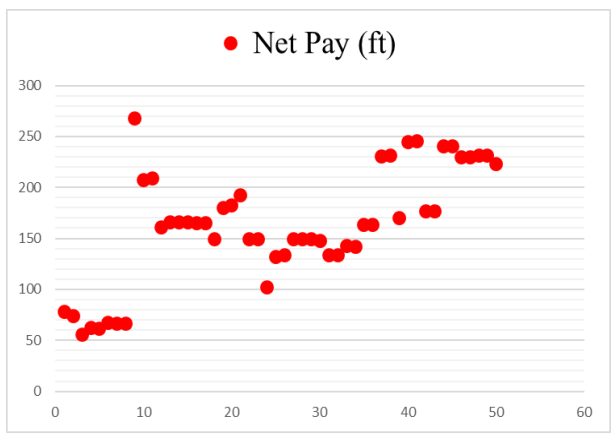
(a) Initial Pressure Estimate (psi) Scatter Plot



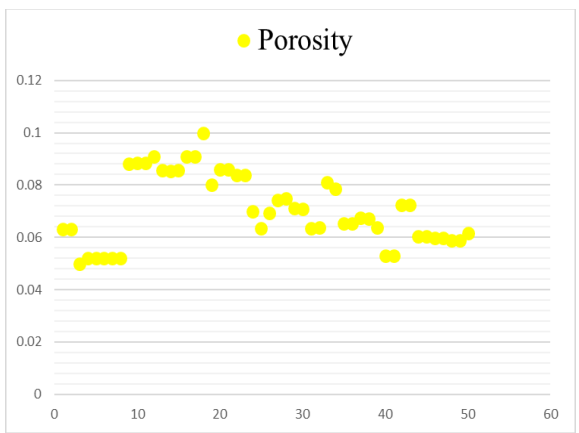
(b) Reservoir Temperature (deg F) Scatter Plot



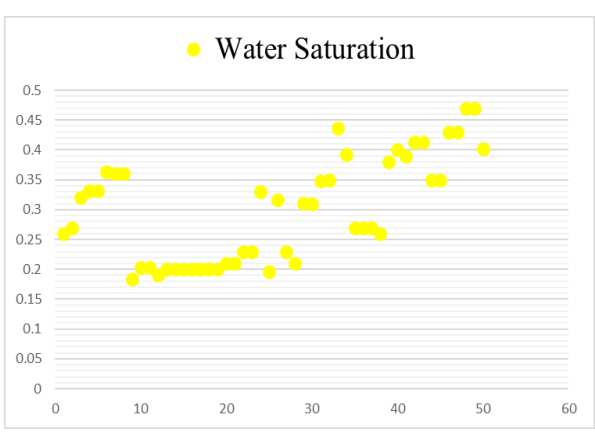
(c) Net Pay (ft) Scatter Plot



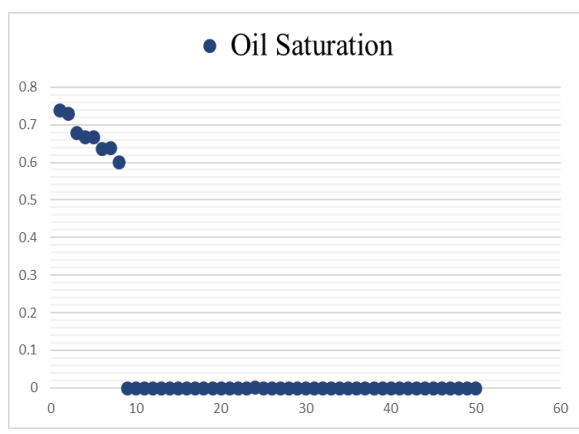
(d) Porosity Scatter Plot



(e) Water Saturation (ft) Scatter Plot



(f) Oil Saturation Scatter Plot



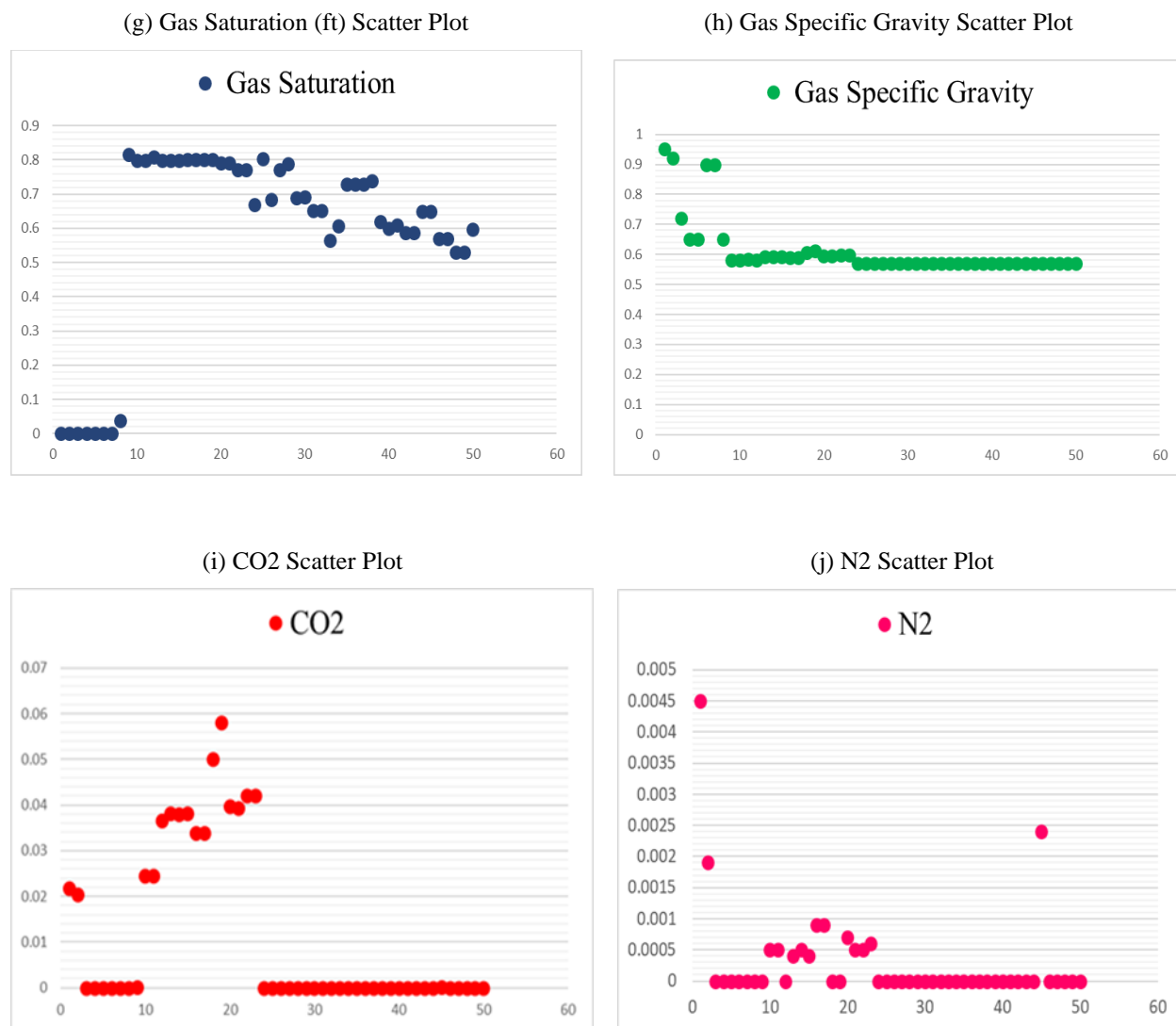
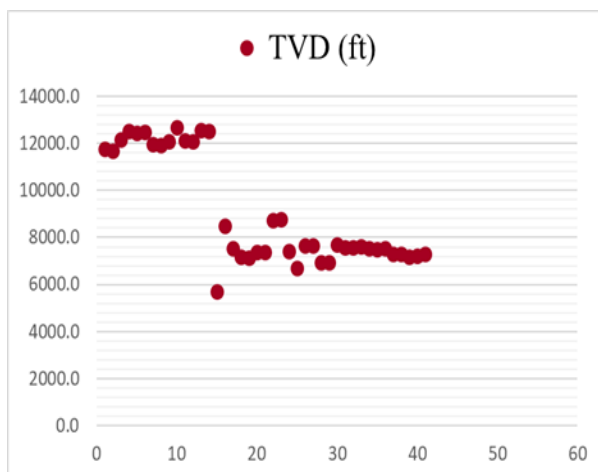


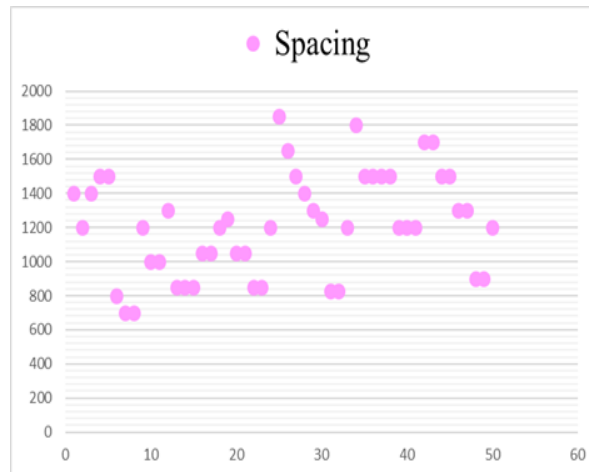
Figure 4.5. Reservoir parameters scatter plots: (a) Initial Pressure Estimate (psi), (b) Reservoir Temperature (F), (c) Net Pay (ft), (d) Porosity (e), Water Saturation, (f) Oil saturation, (g) Gas Saturation, (h) Gas Specific Gravity, (i) CO₂, (j) N₂

Scatterplot for Operational Parameters

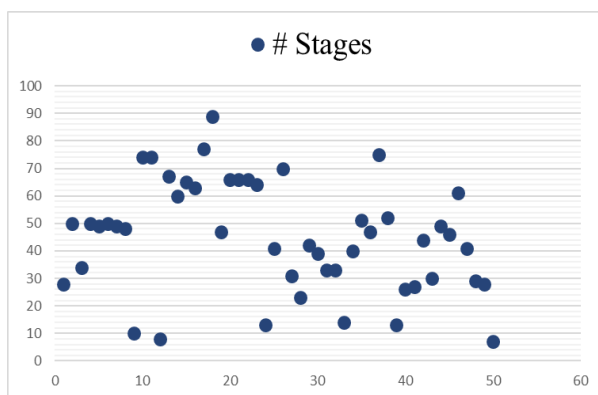
(a) TVD (ft) Scatter Plot



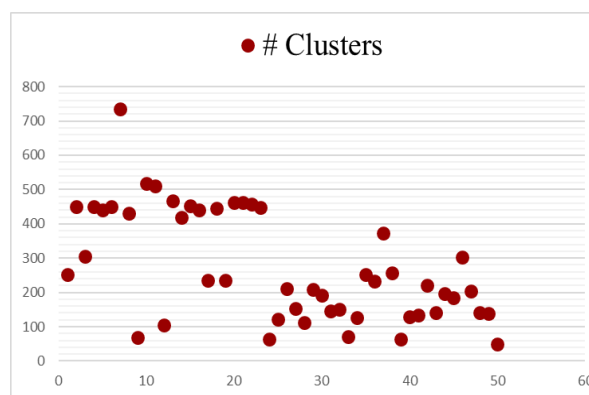
(b) Spacing Scatter Plot



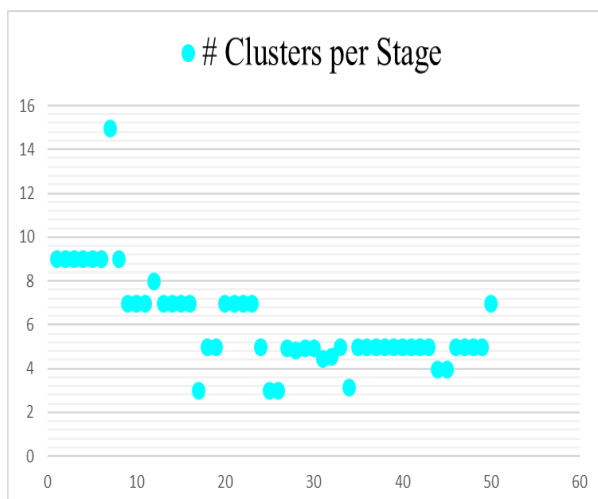
(c) Number of Stages Scatter Plot



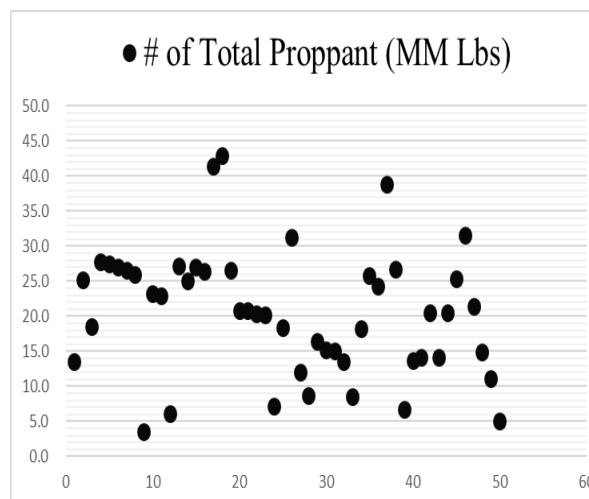
(d) Number of Clusters Scatter Plot



(e) Number of clusters per Stage Scatter Plot



(f) Number of Total Proppant (MM Lbs.) Scatter Plot



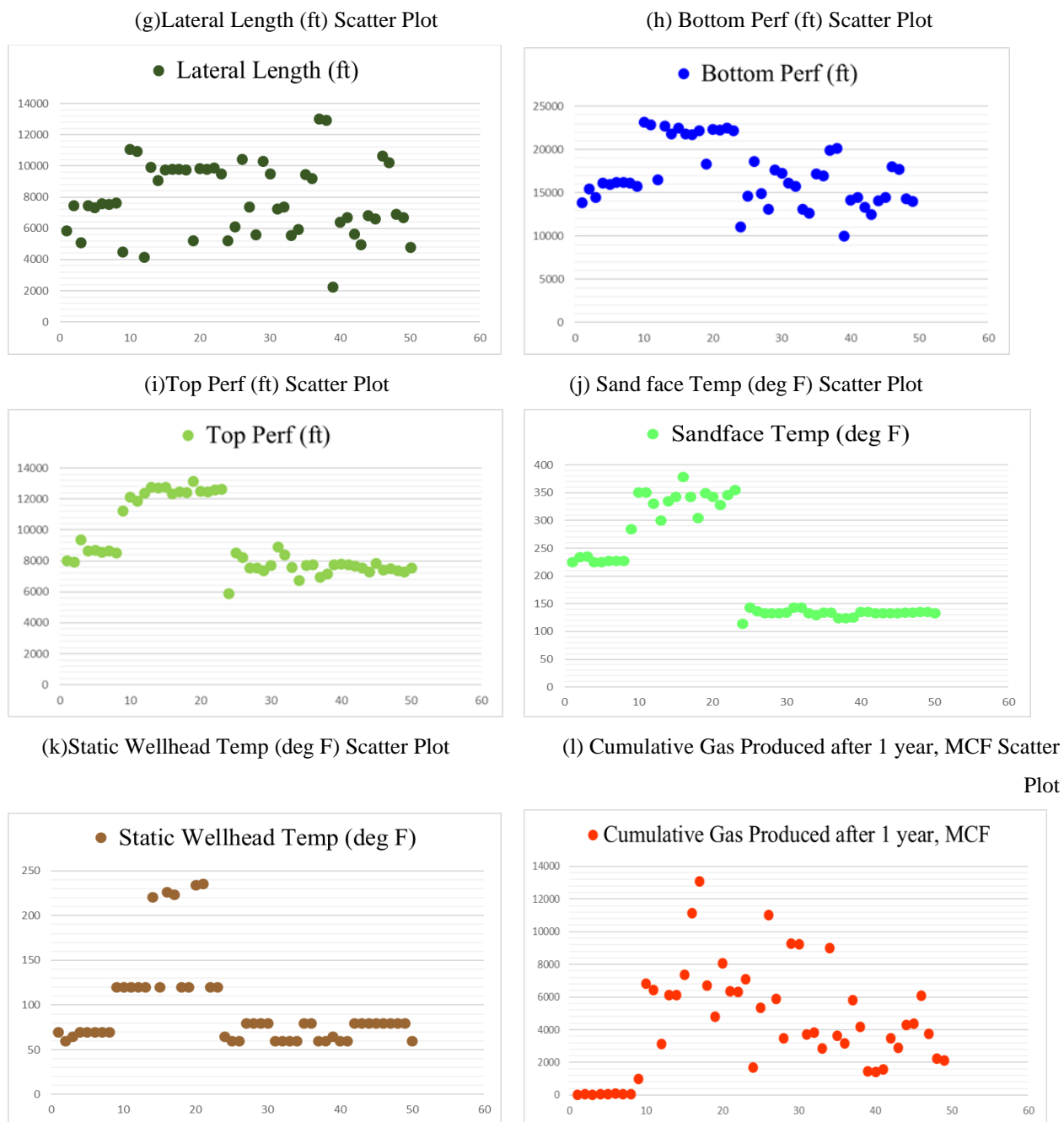


Figure 4.6. Operational parameters histogram plots: (a) TVD (b) Spacing, (c) Number of Stages, (d) Number of clusters, (e) Number of clusters per Stage, (f) Total Proppant (MM Lbs.), (g) Lateral Length (ft.), (h) Top Perf (ft.), (i) Bottom Perf (ft.), (j) Sand face Temp (F), (k) Static wellhead Temp (F), (l) Cumulative Gas Produced after 1 year, MCF.

Multivariate Correlation Plot

In this study, the final step of EDA involved presenting a correlation matrix that builds on the concepts outlined before but now includes all variable pairings, including reservoir and operational characteristics.

Figure 4.7 and 4.8 present the correlation matrix for all variable pairs (dependent and independent).

In Figure 4.7, the highest absolute value of Pearson correlation coefficient is between Top Reservoir temperature (deg F) and Initial pressure estimate (deg F) with a coefficient of 0.93. It can be seen in Figure 4.7 there is a dependency between Reservoir temperature and Initial pressure estimate, CO₂ with Initial Pressure Estimate (psi), CO₂ with Reservoir Temperature (deg F), Gas Specific Gravity with Oil Saturation, Gas Saturation with Net Pay (ft), Gas Saturation with Porosity, Porosity with Reservoir Temperature (deg F) and Cumulative Gas Produced after 1 year, MCF with Gas Saturation.

In Figure 4.8, the highest absolute value of Pearson correlation coefficient is between Top Perf (ft) and TVD (ft) with a coefficient of 1. It can be seen in Figure 4.8 there is a dependency between Bottom Perf (ft) and Stages, Bottom Perf (ft) with Lateral Length (ft), Lateral Length (ft) with Stages, Number of Cluster per stage with Number of Clusters, and Bottom Perf (ft) with Number of Clusters.

The reason for noticing the preceding outlier points was the dependence between independent variables (Predictors), as well as between dependent variables (Response) and independent variables the outlier points cannot be the result of an inaccurate input value into the dataset since this dataset was created via numerical simulation scenarios. As a result, this dependency results in additional log normality, which is evident in the box plots and histograms of these variables.

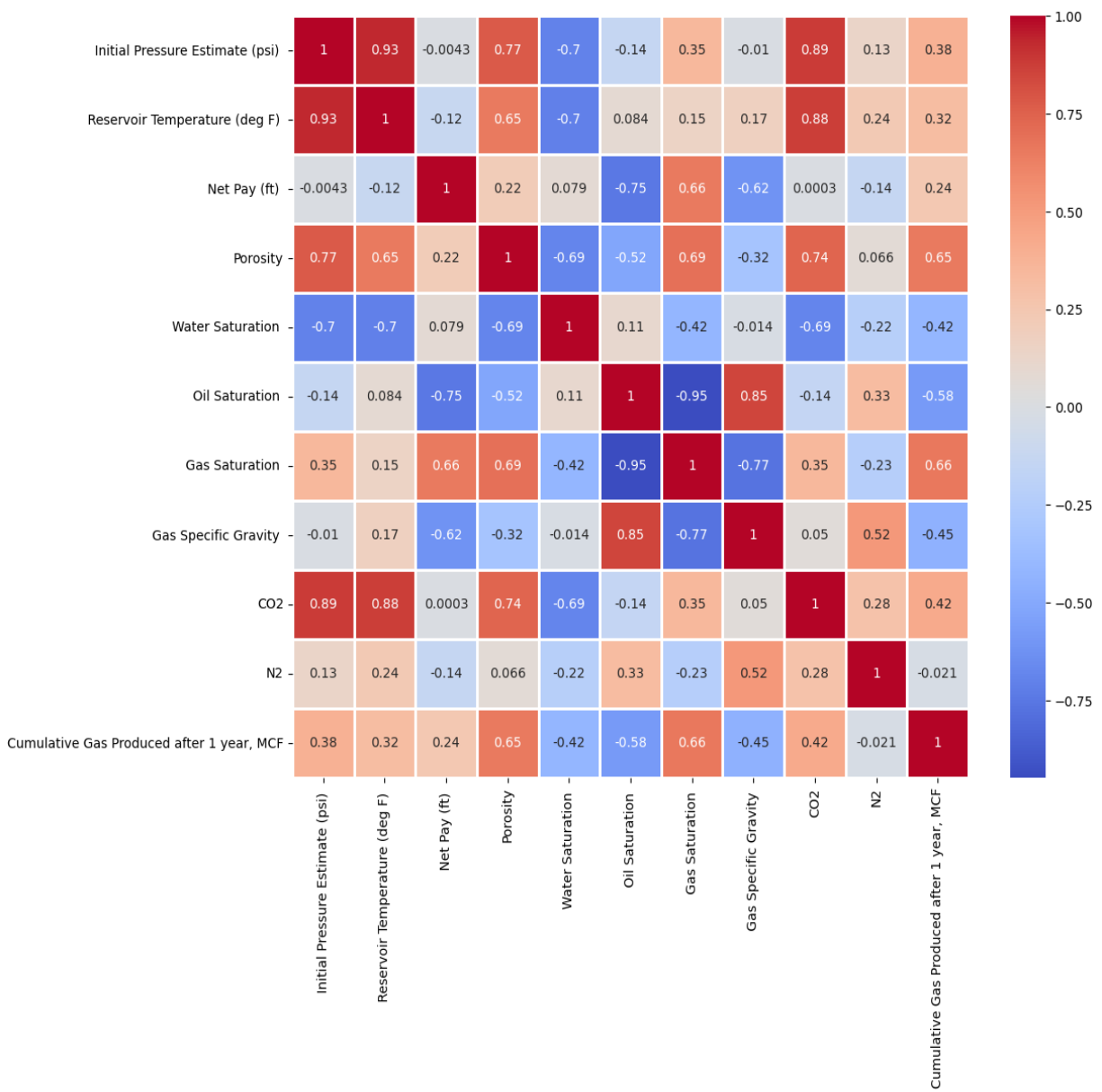


Figure 4.7. Multivariable correlation plot for reservoir parameters: Initial Pressure Estimate (psi), Reservoir Temperature (F), Net Pay (ft), Porosity, Water Saturation, Oil saturation, Gas Saturation, Gas Specific Gravity, CO2 ,N2 and Cumulative Gas Produced after 1 year, MCF

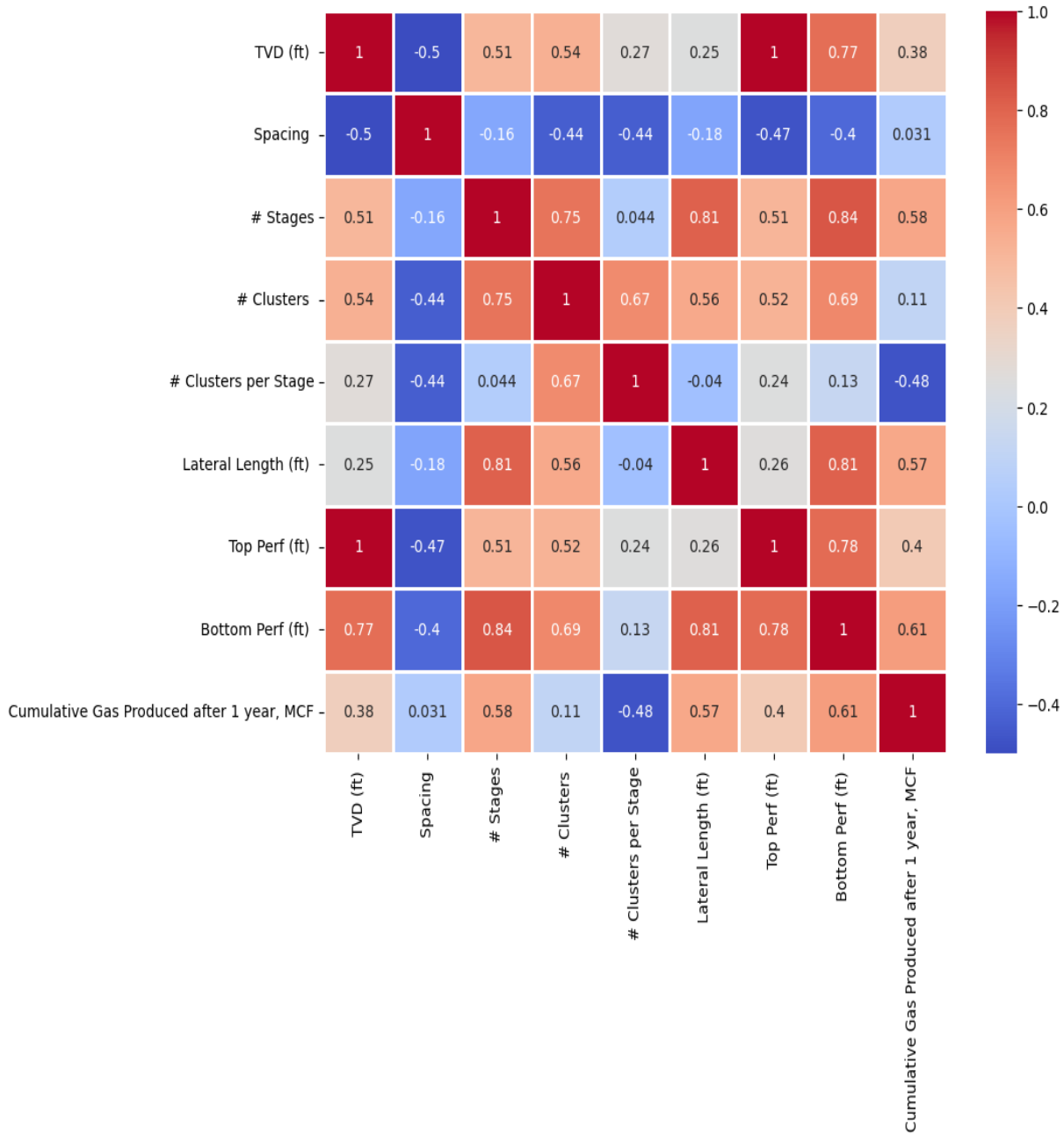


Figure 4.8. Multivariable correlation plot for operational parameters: TVD, Spacing, Stages, Number of clusters, Number of clusters per Stage, Total Proppant (MM Lbs.), Lateral Length (ft.), Top Perf (ft.), Bottom Perf (ft.), n Cumulative Gas Produced after 1 year, MCF

Predictive modeling

The EDA method utilized in the aforementioned section is a crucial methodology employed in this study to verify the variables that have a relationship with the cumulative gas generated after a year, MCF, as well as figuring out patterns and trends to carry out predictive modeling. The decision tree and random forest approaches were used in this study to predict the cumulative gas produced after a year, or MCF. These predictive models are crucial for giving precise forecasts of the total amount of gas generated after a year, measured in MCF, given the dataset at hand. (Hastie, T., Tibshirani, R., & Friedman, 2008)

Decision Tree

This method of evaluating the decision tree model's level of accuracy is used since it is simple and uncomplicated as shown in Figure 4.9 to 4.17. These techniques aid in lowering the variance of a statistical-machine learning algorithm and enhancing the performance of these techniques, as was previously discussed in the methodology chapter. To evaluate the prediction error and determine whether there has been an improvement, three plots were created.

As R^2 is an indicator of regression error that supports the model's effectiveness. The amount by which the independent variables can adequately characterize the value of the response or target variable is what it represents. Always between 0 and 1 (0% to 100%), that is the range for R-Square's value. The linear regression function line is close to many data points when the R-Squared value is high. When the linear regression function line has a low R-Squared value, the data are not well fitted by the function line.

In Table 4.2 we can see that by training the model with 70% of training and 30% of testing set it gives the highest value of R^2 value, it means that many data points are close to the linear regression function line, comparative to other training and testing tests used in decision tree in this thesis.

Furthermore, it can be observed in Figure 4.11, 4.14, 4.17 that bottom perf, (ft) is the most influential predictor and has an immense impact on the performance of shale well followed by cluster per stage.

Table 4.2. Training and testing sets for decision tree model

	R^2	R
Training 70%	96%	98%
Testing 30%	70.6%	84%
Training 75%	95.5%	97.7%
Testing 25%	56.8%	75%
Training 80%	95.7%	97.8%
Testing 20%	57.7%	76%

The model prediction error can be estimated by a plot of actual and predicted values from the decision tree with 70% training and 30% testing set. This corresponds to a prediction error of:

MAE: 1.5×10^3 MCF

MSE: 6.7×10^6 MCF

RMSE: 2.5×10^3 MCF

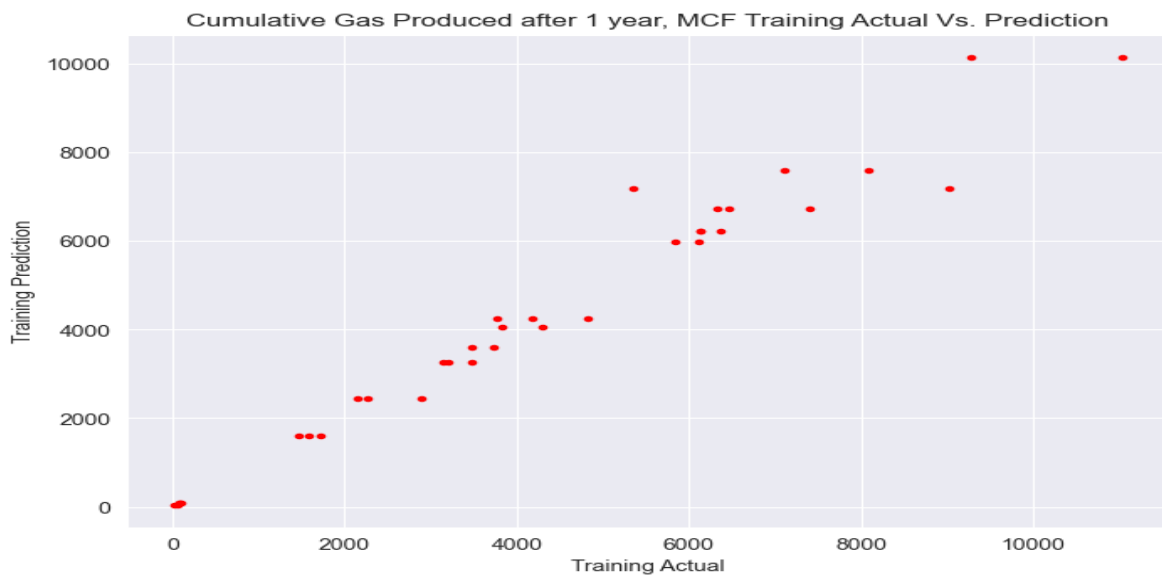


Figure 4.9. Training actual vs prediction using decision tree (70% training).

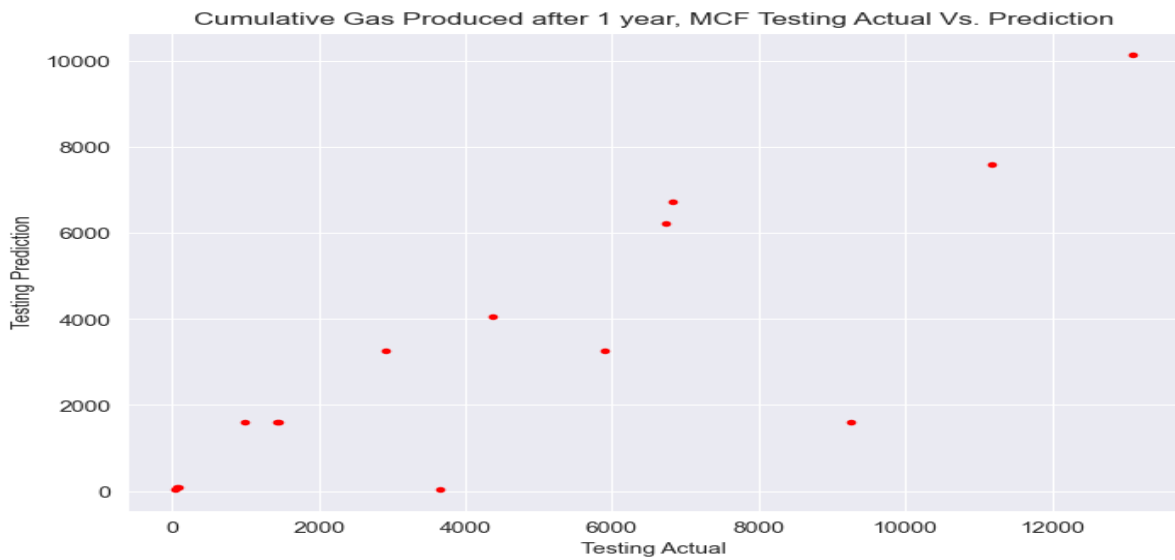


Figure 4.10. Testing actual vs prediction using decision tree (30% training).

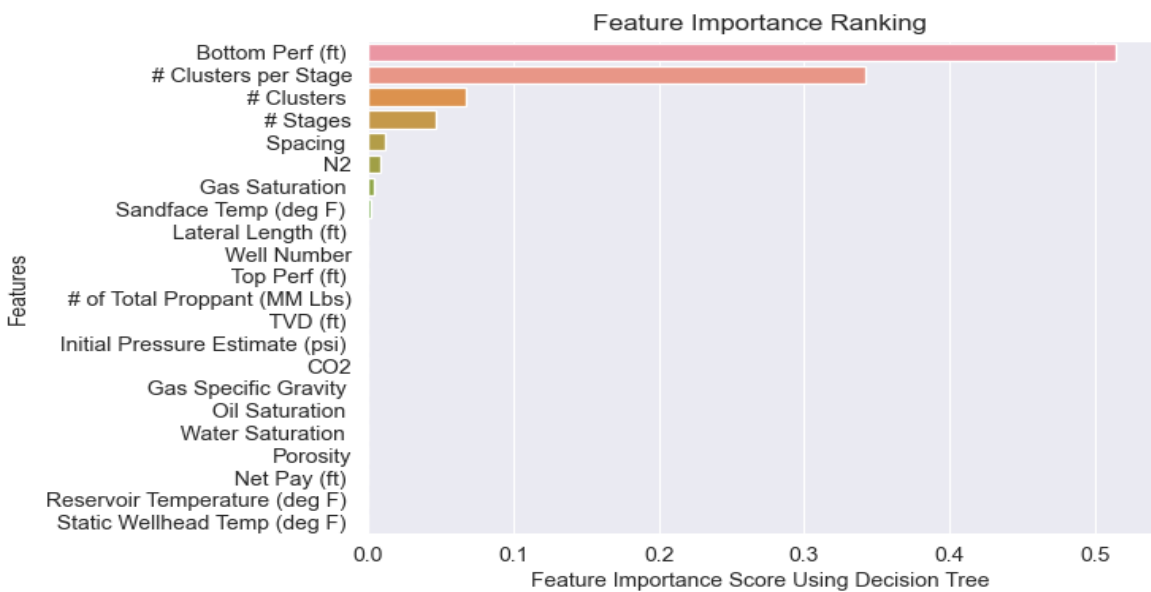


Figure 4.11. Feature importance score using decision tree (70/30).

The second plot of decision tree with 75% training and 25% testing set can be seen in Figure 4.12 and 4.13 this technique provided the following prediction error:

MAE: 1.5×10^3 MCF

MSE: 7.1×10^6 MCF

RMSE: 2.6×10^3 MCF

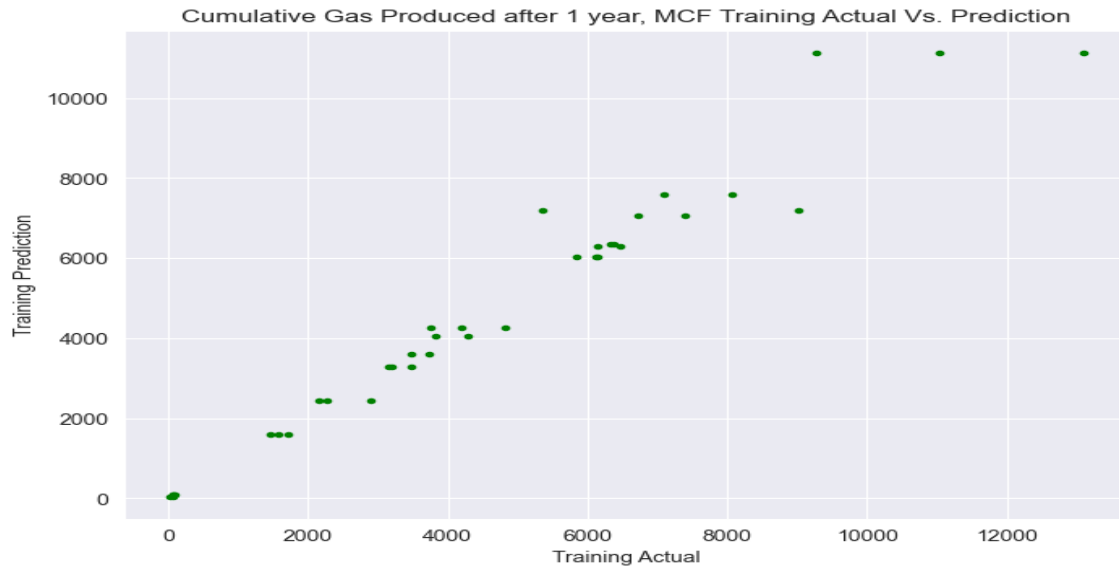


Figure 4.12. Training actual vs prediction using decision tree (75% training).

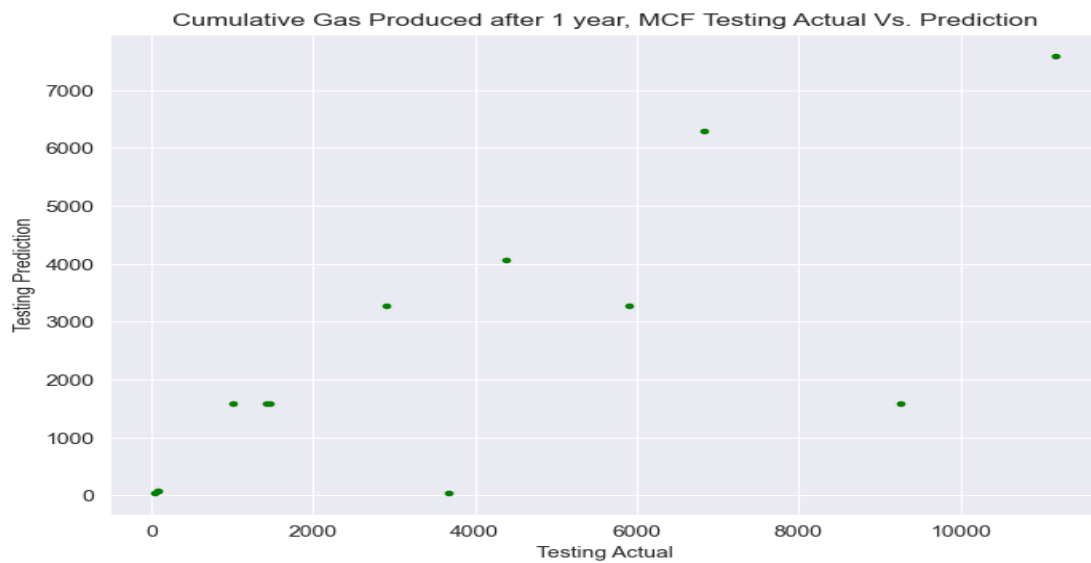


Figure 4.13. Testing actual vs prediction using decision tree (25% training).

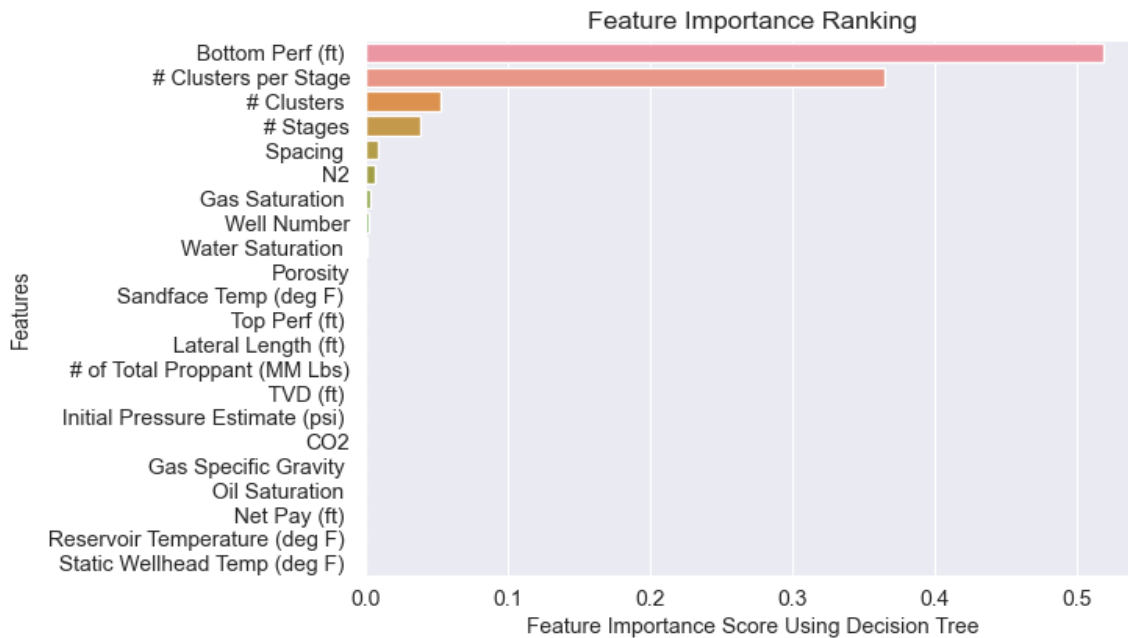


Figure 4.14. Feature importance score using decision tree (75/25).

The third and final plot of decision tree with 80% training and 20% testing test can be seen in Figure 4.15, and 4.16 this technique produced the following prediction error:

MAE: 1.8×10^3 MCF

MSE: 9.2×10^6 MCF

RMSE: 3×10^3 MCF

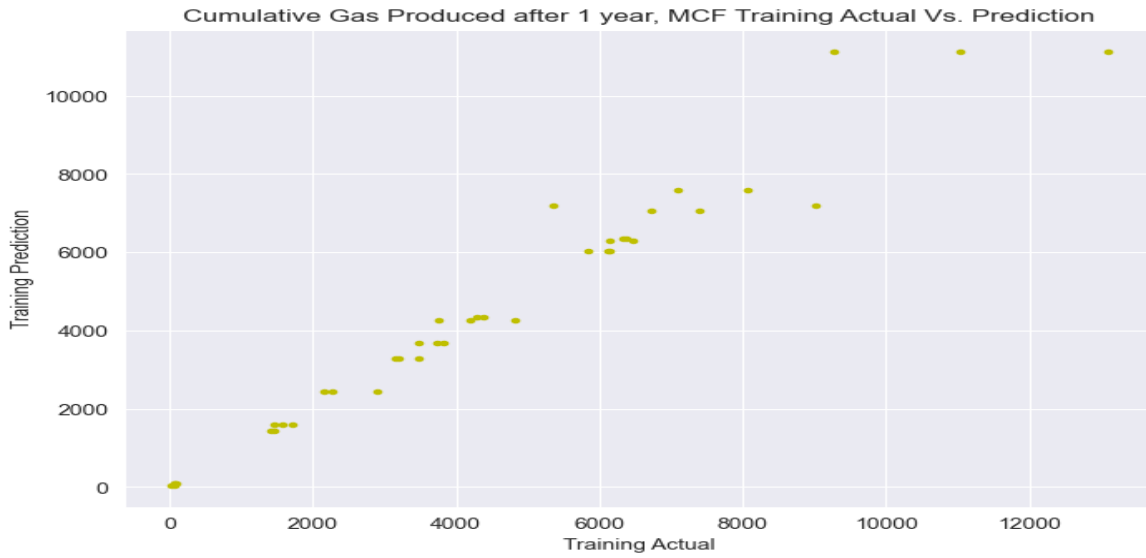


Figure 4.15. Training actual vs prediction using decision tree (80% training).

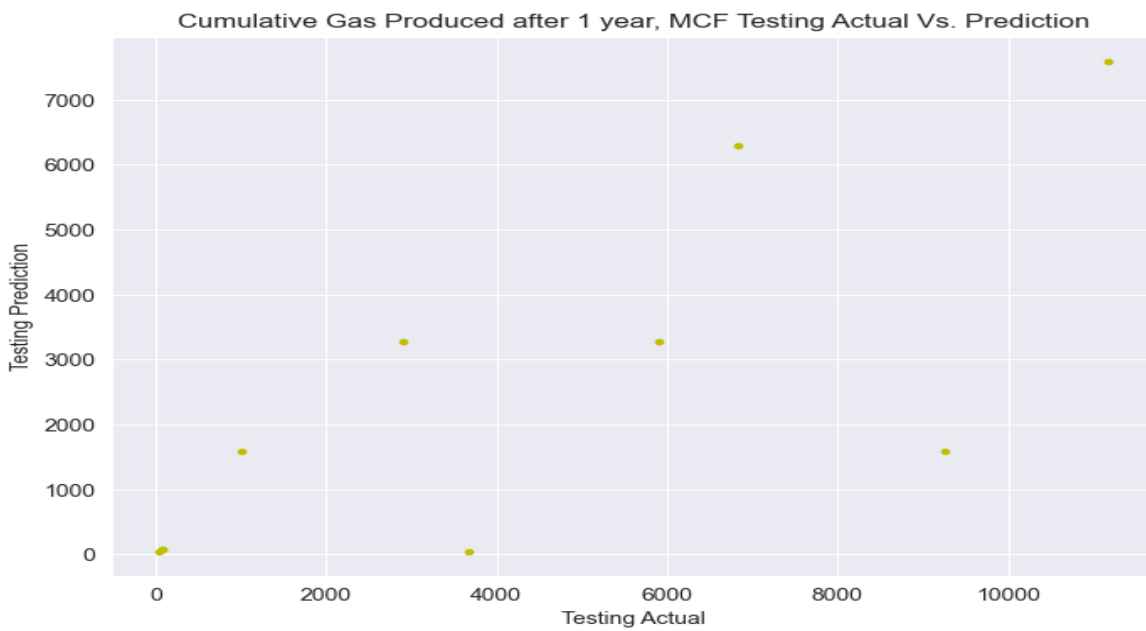


Figure 4.16. Testing actual vs prediction using decision tree (20% training).

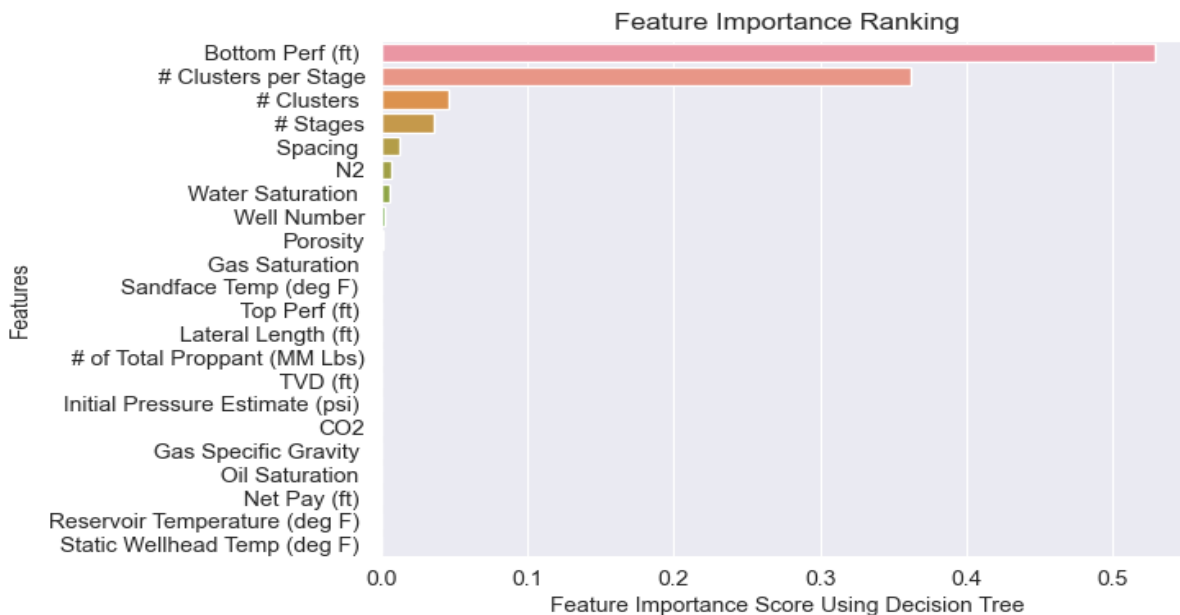


Figure 4.17. Feature importance score using decision tree (80/20).

Random Forest

The accuracy (R^2) of the training and testing sets is displayed in Table 4.3. As can be seen, using 70% training and 30% testing test results in an 84.3% testing rate compared to 70.6% of the decision tree, using 75% training and 25% testing test results in an 87.3% testing rate compared to 56.8% of the decision tree, and using 80% training and 20% testing test results in an 87.7% testing rate compared to 57.7% of the decision tree.

Therefore, it appears that the random forest algorithm outperforms the decision tree without further parameter fine-tuning. In figures 4.18 and 4.19, 4.21 and 4.22, 4.24 and 4.25, the plots of actual versus predicted training and testing data are visualized. As shown, compared to the decision tree model, MAE, MSE, and RMSE values are lower.

The significant properties obtained by random forest are distinct from those obtained by decision tree, as shown in Figures 4.20, 4.23, and 4.26. This is mostly attributable to the random forest model's greater accuracy. It is advised to use the random forest model, as it has a higher level of accuracy.

As was previously said, the goal in classification issues is to reduce the Gini impurity (assuming Gini impurity is chosen) Eq. 3.7. As a result, the nodes that result in the greatest decrease in Gini impurity are found at the beginning of the trees, whilst the nodes that result in the least reduction are found towards the end of the trees. Feature ranking is carried out by tree-based algorithms in this manner.

Table 4.3. *Training and testing sets for random forest models*

	R^2	R
Training 70%	94%	96.9%
Testing 30%	84.3%	91.8%
Training 75%	95.2%	97.6%
Testing 25%	87.3%	93.4%
Training 80%	95.3%	97.6%
Testing 20%	87.7%	93.6%

The model prediction error can be estimated by a plot of actual and predicted values from the random forest with 70% training and 30% testing set. This corresponds to a prediction error of:

MAE: 1.4×10^3 MCF

MSE: 4.7×10^6 MCF

RMSE: 2.1×10^3 MCF

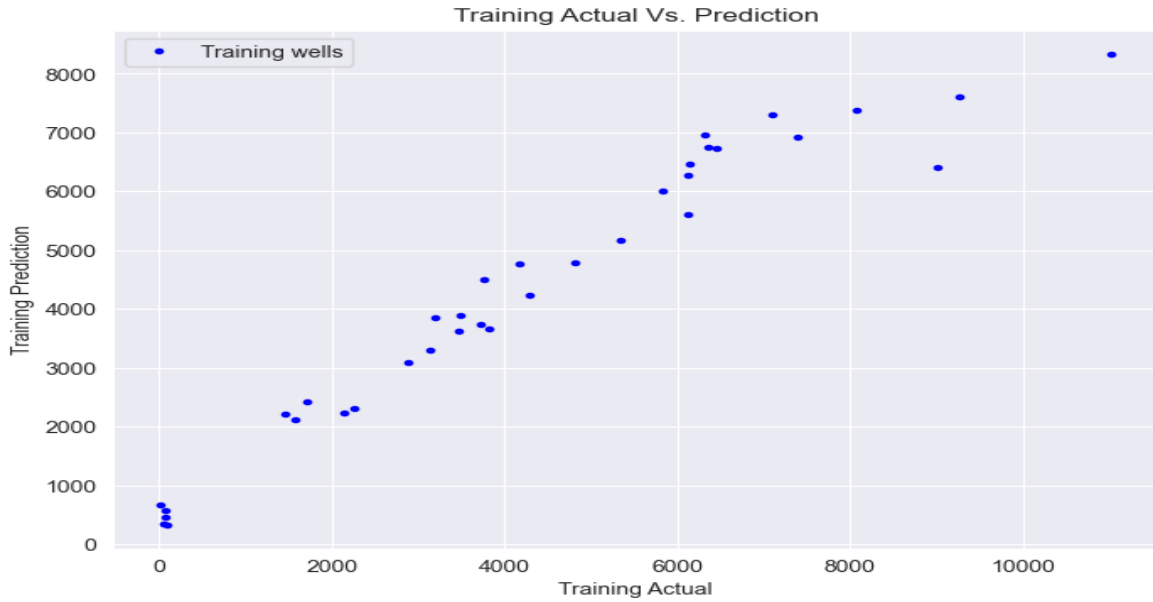


Figure 4.18. Training actual vs prediction using random forest (70% training).

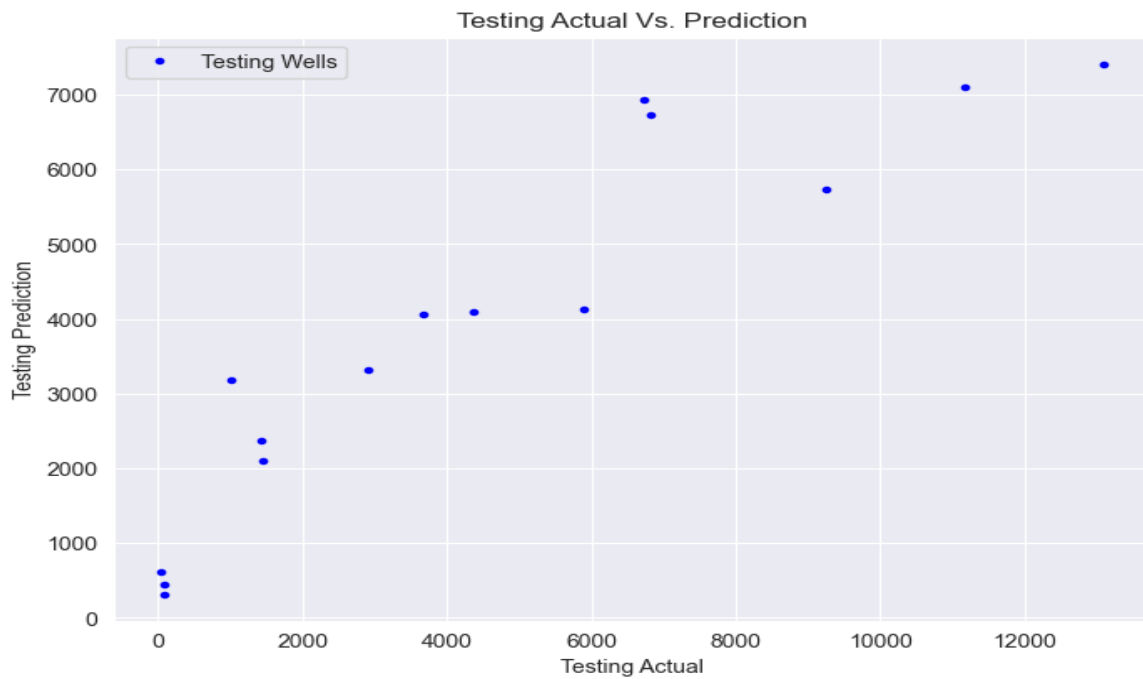


Figure 4.19. Testing actual vs prediction using random forest (30% training).

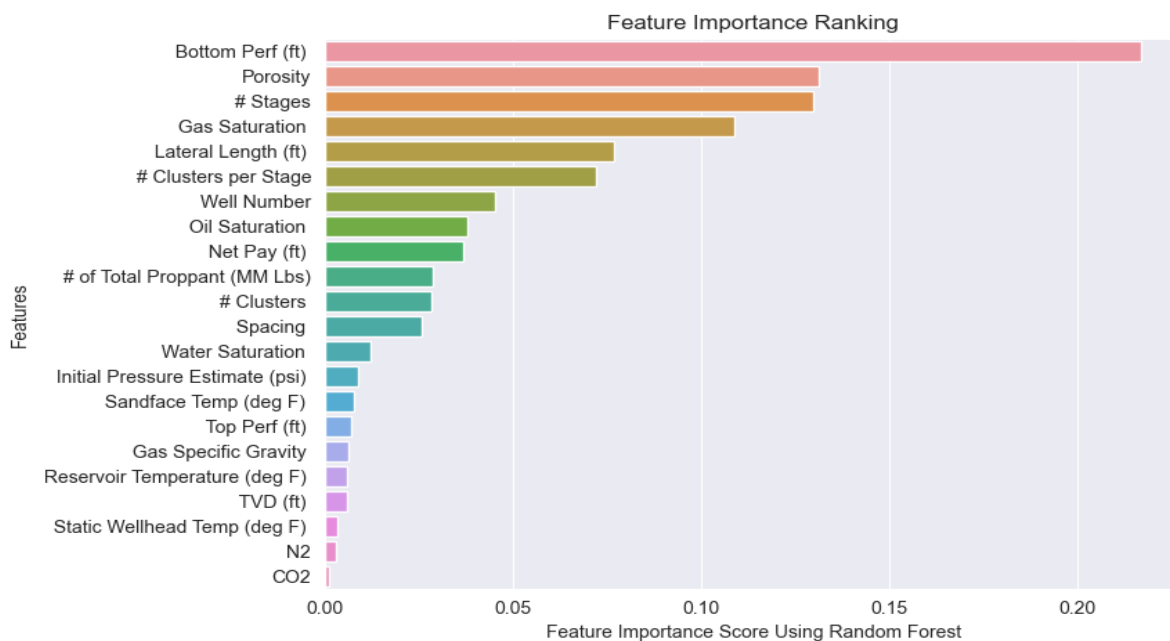


Figure 4.20. Feature importance score using random forest (70/30).

The second plot of random forest with 75% training and 25% testing set can be seen in Figure 4.21 and 4.22 this technique provided the following prediction error:

MAE: 1.1×10^3 MCF

MSE: 2.7×10^6 MCF

RMSE: 1.6×10^3 MCF

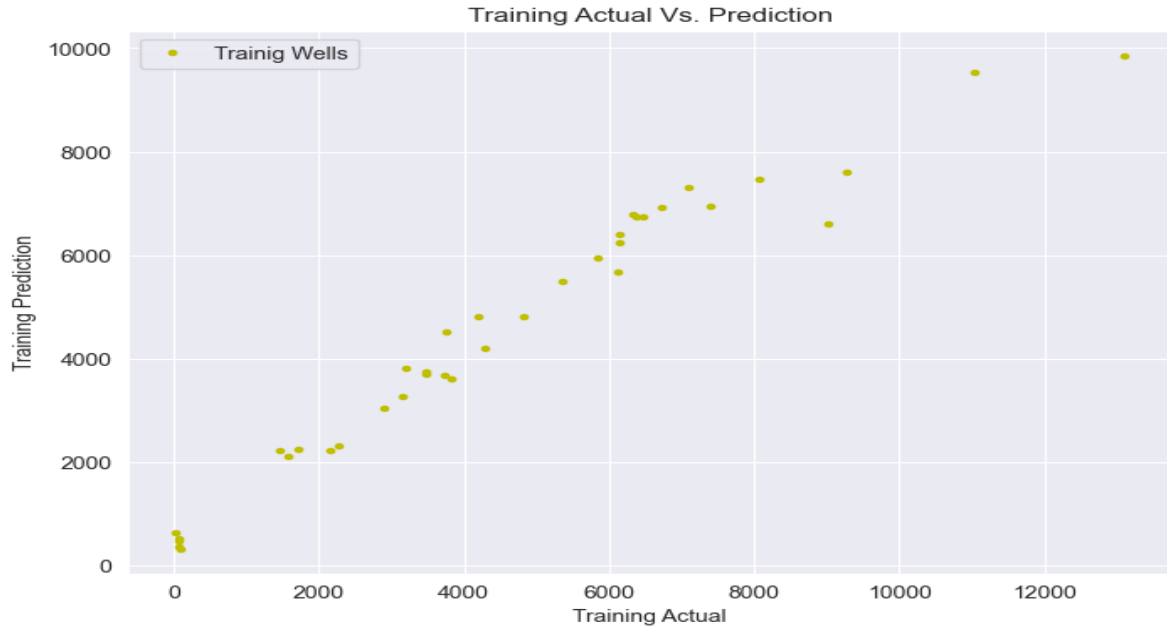


Figure 4.21. Training actual vs prediction using random forest (75% training).

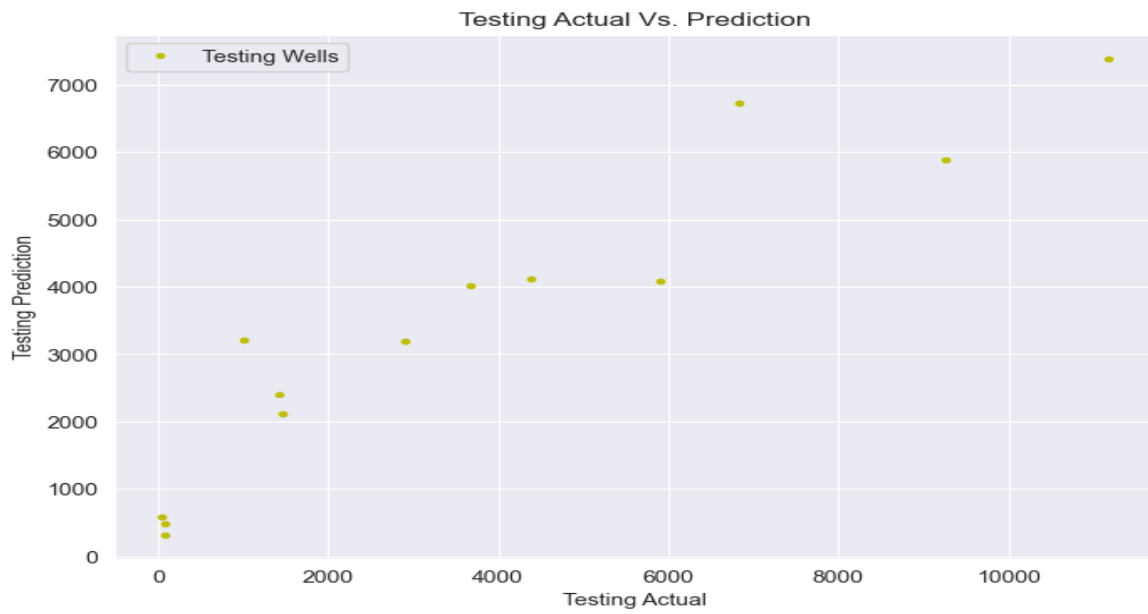


Figure 4.22. Testing actual vs prediction using random forest (25% training).

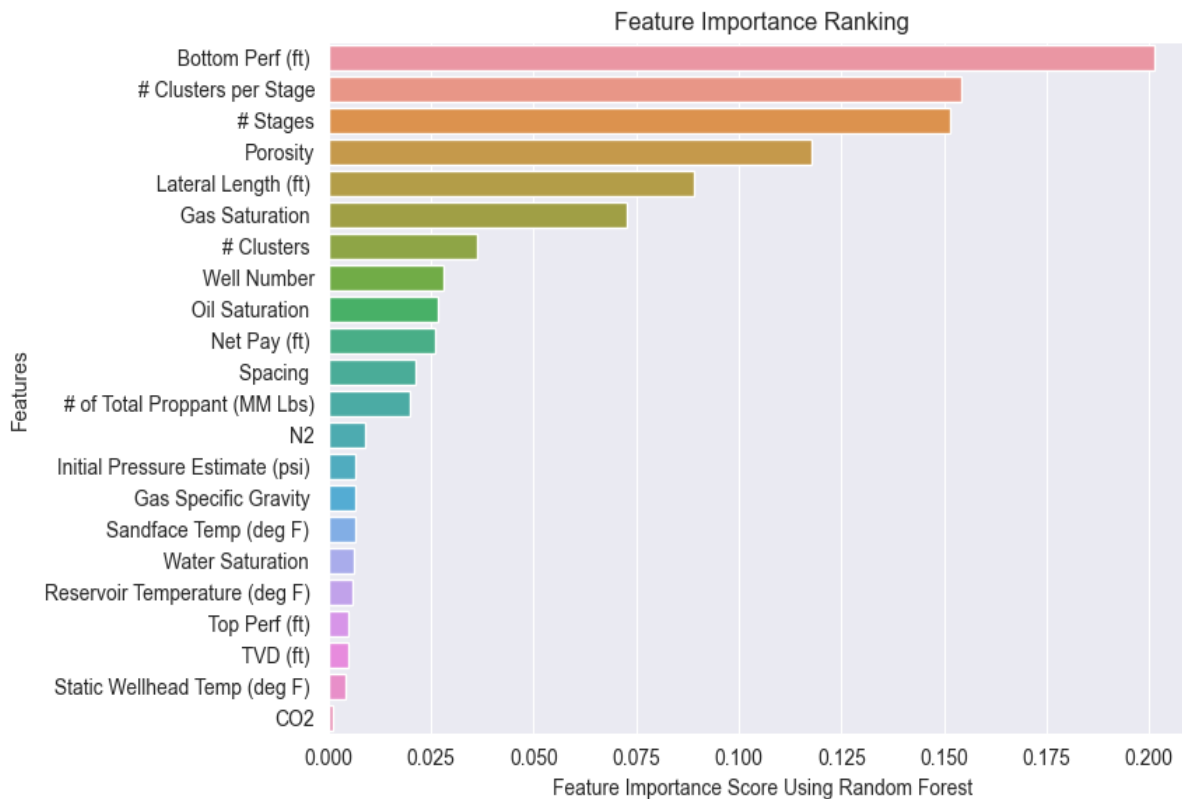


Figure 4.23. Feature importance score using random forest (75/25).

The third and final plot of random forest with 80% training and 20% testing test can be seen in Figure 4.24 and 4.25 this technique produced the following prediction error:

MAE: 1.2×10^3 MCF

MSE: 3.4×10^6 MCF

RMSE: 1.8×10^3 MCF

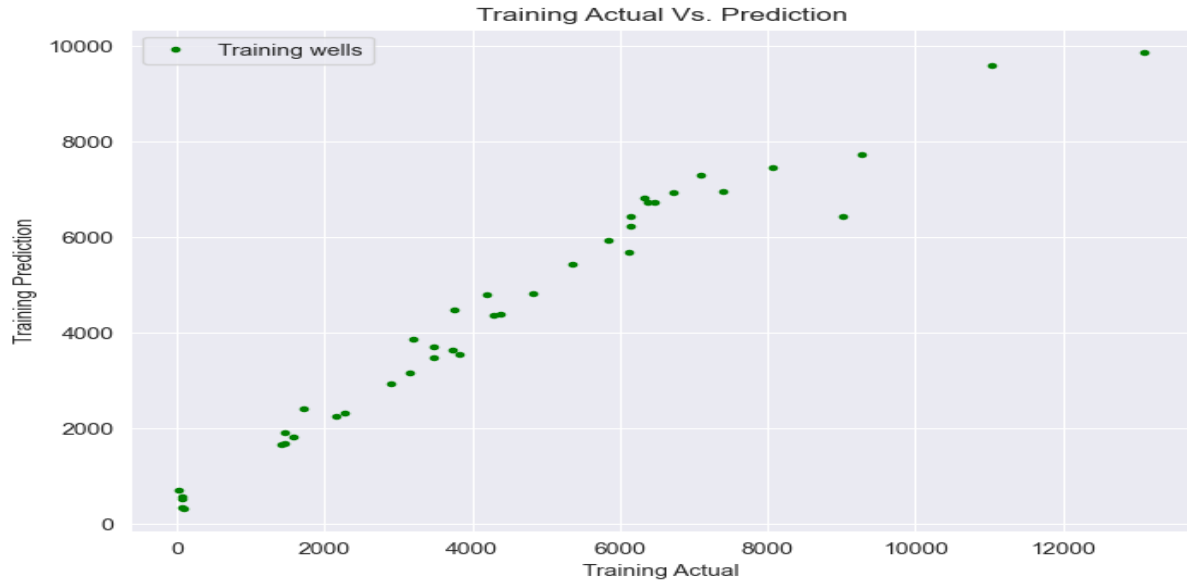


Figure 4.24. Training actual vs prediction using random forest (80% training).

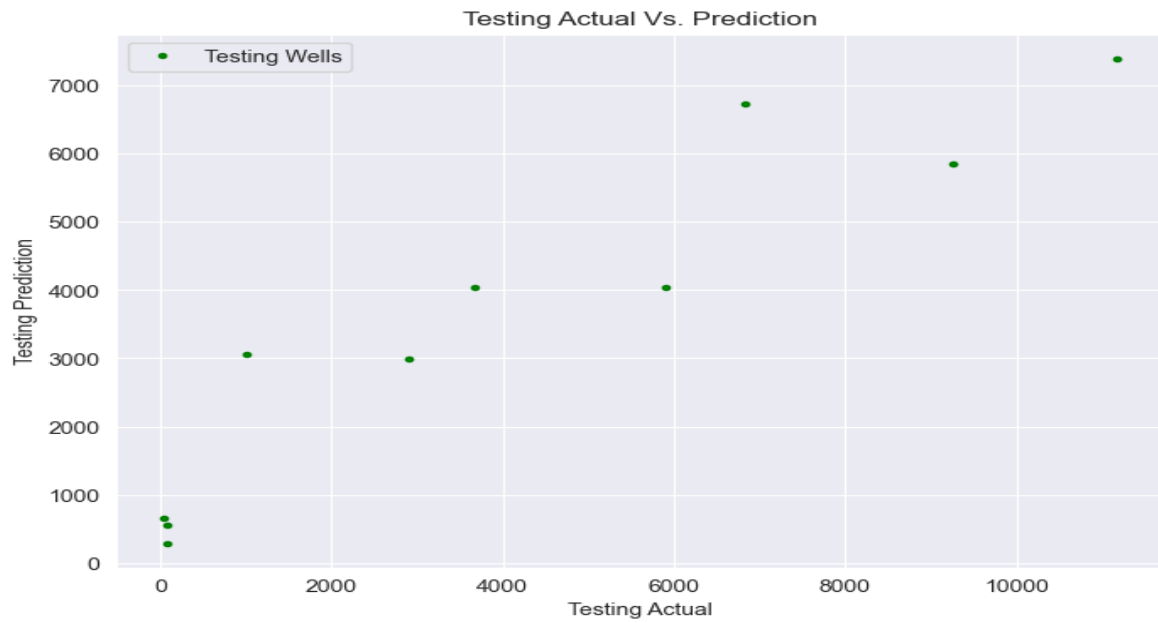


Figure 4.25. Testing actual vs prediction using random forest (20% training).

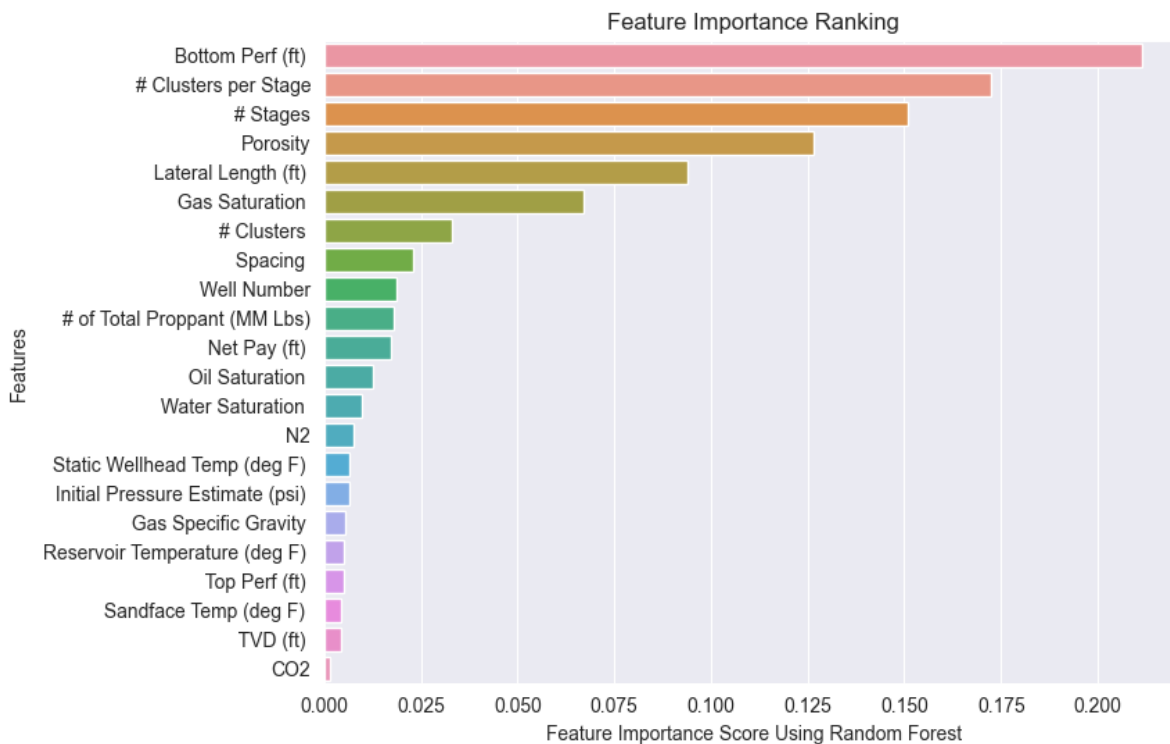


Figure 4.26. Feature importance score using random forest (80/20).

Comparison of Decision Tree and Random Forest Using Bar Chart Plot

In this review, by creating model by training test we can visualize that Decision Tree (DT) and Random Forest (RF) have approximately the same values. And by regulating the model by testing set Random Forest algorithms have better accuracy rates than Decision Tree algorithms.

In comparison to the Decision tree method, the random Forest algorithm have a greater accuracy of about 84%, 87%, and 87%.

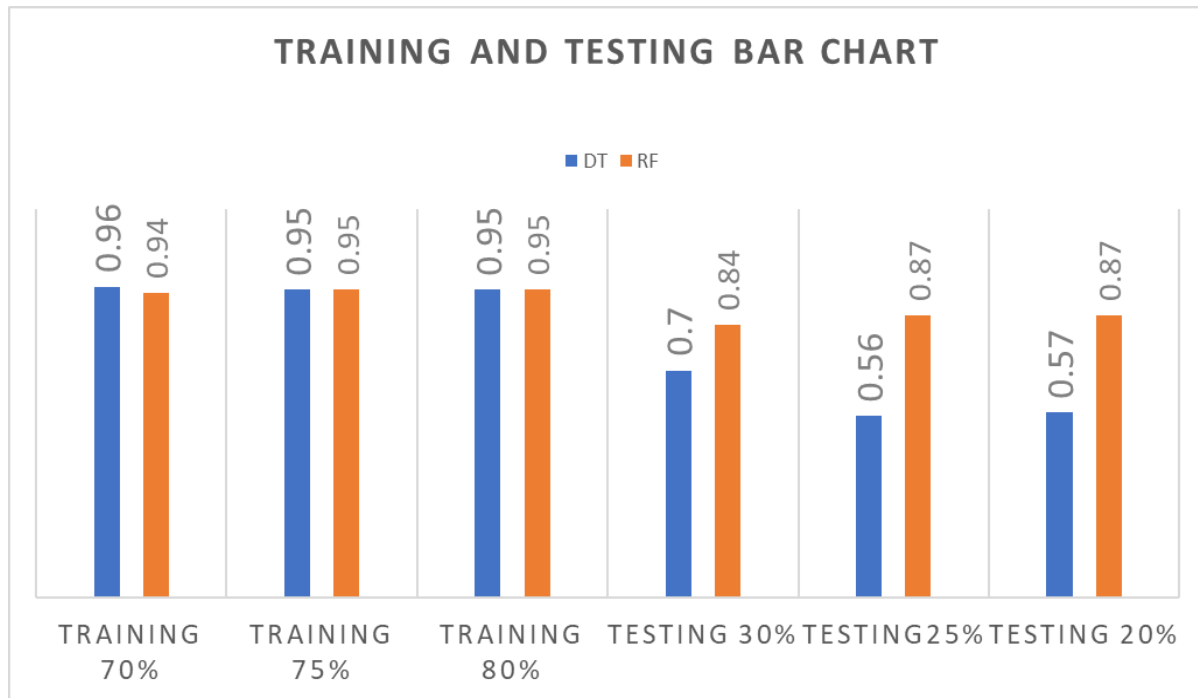


Figure 4.27. Decision tree and random forest bar chart comparison of training and testing tests.

Variable Importance

By predicting the cumulative gas output after a year, this study's last section identified the performance forecasting of shale wells in unconventional shale-gas reservoirs. The key way to control this process is to examine the response variable in the context of a large number of predictor factors. DT and RF contain built-in functions for carrying out such a procedure to find the most significant predictors, which can be used to accomplish this. For the DT and RF models, the percent decrease in RMSE is used to determine a relevance of predictor. Let's compare how important each feature is according to DT and RF.

I. Decision Tree

According to the decision tree's feature relevance rankings, Bottom Per(ft) is the most significant predictor and has a significant impact on shale well performance forecasts, followed by Clusters per stage, Clusters, and stages.

II. Random Forest

Bottom Perf (ft.), followed by Clusters per stage, Stage, Porosity, and Lateral Length (ft.), is the most influential predictor and has a significant impact on the performance

forecasting of shale wells, as can be seen from the feature importance rankings of random forest.

Comparison of Feature Importance Rankings of DT and RF

It should be observed that the top two decisive predictors (Bottom Perf (ft.) and Clusters per stage) for the shale-gas reservoirs for both the DT and RF models are identical with 20 and 25% of testing tests, and the top one for all of the feature important are identical.

The other rating for the predictor, on the other hand, is different because, for instance, RF ranks the most significant predictors differently from DT. It is important to talk about the key finding of the parameters in terms of their physical meaning.

We will evaluate the importance of the high-performance predictors in this study to determine whether they are physically logical.

- **Bottom Perf (ft):** When it comes to hydraulic fracturing and horizontal wells, a perforation is crucial. It is forming a conduit between the pay zone and the wellbore to make it easier for gas to flow there. At the perforations at the bottom of a well, the pressure of the liquid inside the wellbore causes the rock to fracture.
- **Clusters Per Stage:** Since there are fewer entrance opportunities for hydraulic fracturing if there are fewer clusters each stage. Greater surface area near the wellbore each stage is produced by more clusters, which can maximize the early gas recovery.
- **Porosity:** Shale are distinguished by having extremely little porosity. a kind of secondary porosity brought about by the rock's tectonic fractures. Fractures normally do not have much volume on their own, but by linking preexisting pores, they greatly increase permeability.
- **Lateral Length (ft.):** The most important aspect in determining the production and financial advantages of horizontal wells is lateral length. The horizontal wellbore length is essential because the well crosses highly conductive cracks, which would facilitate the extraction of shale gas for the purpose of injecting CO₂. Additionally, a long horizontal wellbore length would increase the area in contact with the fracture permeability zone, which would obviously affect the well's productivity index.

CHAPTER V

Concluding Remarks

Data analytics is employed in this study to determine the key factors that influence the total gas generated after a year MCF. In this study, unconventional shale reservoir is the main topic. In order to comprehend the features and patterns inside a dataset in shale well reservoirs, an EDA using data mining and visualization was carried out.

Predictive models were created using statistical and machine-learning techniques after gaining insights into the information. Assessing the link between reservoir parameters and operational factors in order to precisely forecast process performance. Then, the predictive effectiveness of each of these models was evaluated to determine which model had the highest accuracy in estimating the total gas generated after 1-year MCF.

Conclusions

The major conclusions from this study are as follows:

- 1) In unconventional shale reservoirs, operational parameter is increasingly important for shale well performance. The most important indicator is Bottom Perf(ft), has a significant impact on shale well performance forecasting.
- 2) The most influential parameters of performance forecasting of shale well according to DT are:
 - Bottom Perf(ft)
 - Clusters per stage
- 3) The most influential parameters of performance forecasting of shale well according to RF are:
 - Bottom Perf(ft)
 - Cluster per Stage
 - Number of stages
 - Porosity
 - Lateral Length(ft)

- 4) The maximum proportion of variation is explained with a R^2 value when using Random Forests (RF), which also has the lowest prediction error when compared to Decision Tree. This outcome supports the literature's assertion that one of the most potent machine learning methods is the RF model.
- 5) Regression tree (RF) can rank the most important factors that affect the total amount of gas produced and are simple to understand.
- 6) The reliance between the predictors and response variables, which adds more log normality and manifests as the outlier points, is the root cause of the EDA outlier points. There is no way to link these outlier points to erroneous input values in the dataset.

Recommendations

- To improve the prediction capabilities of the machine learning model, more reservoir information should be provided. The actual reservoir dataset can help to make the decision tree and random forest (machine learning model) more applicable.
- The problem can be made simpler by converting the regression tree into a classification tree, which can then be used to forecast whether performance will be high or low.

References

- Boosari, S. S. H., Aybar, U., & Eshkalak, M. O. (2015). Carbon Dioxide Storage and Sequestration in Unconventional Shale Reservoirs. *Journal of Geoscience and Environment Protection*, 03(01), 7–15. <https://doi.org/10.4236/gep.2015.31002>
- Breiman, L. (1994). Bagging predictors. Department of Statistics, University of California at Berkeley. <https://www.stat.berkeley.edu/wbreiman/bagging.pdf>.
- Breiman, L. (2011). Random forests. Springer. <https://doi.org/10.1023/A:1010933404324>. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Cornell University. <https://doi.org/10.1145/2939672.2939785>.
- Ertekin, T., Abou-Kassem, J. H., & King, G. R. (2001). Basic Applied Reservoir Simulation. <https://store.spe.org/Basic-Applied-Reservoir-Simulation--P12.aspx>
- Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). <https://web.stanford.edu/~hastie/Papers/ESLII.pdf>
- Holdaway, K. R. (2009). Exploratory Data Analysis in Reservoir Characterization Projects. *Society of Petroleum Engineers*, 1(October), 1–20. <https://doi.org/10.3997/2214-4609-pdb.170.spe125368>
- Kam Ho, T. (1995). Random decision forests (pp. 278e282). <https://web.archive.org/web/20160417030218/http://ect.belllabs.com/who/tkh/publications/papers/odt.pdf>.
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. In Springer (Vol. 26). http://appliedpredictivemodeling.com/s/Applied_Predictive_Modeling_in_R.pdf
- Lolon, E., Hamidieh, K., Weijers, L., Mayerhofer, M., Melcher, H., & Oduba, O. (2016). Evaluating the Relationship Between Well Parameters and Production using Multivariate Statistical Models: A Middle Bakken and Three Forks Case History. *Society of Petroleum Engineers - SPE Hydraulic Fracturing Technology Conference, HFTC 2016*. <https://doi.org/10.2118/179171-ms>
- Mishra, S., & Datta-Gupta, A. (2018). *Applied Statistical Modeling and Data Analytics: A Practical Guide for the Petroleum Geosciences*. Candice Janco.

- <https://www.elsevier.com/books/applied-statistical-modeling-data-analytics/Mishra/978-0-12-803279-4>
- Mishra, S., & Lin, L. (2017). Application of Data Analytics for Production Optimization in Unconventional Reservoirs: A Critical Review. SPE/AAPG/SEG Unconventional Resources, Technology Conference 2017, c, 1060–1065. <https://doi.org/10.15530/urtec2017-2670157>
- Mishra, S., Oruganti, Y. D., & Sminchak, J. (2014). Parametric analysis of CO₂ geologic sequestration in closed volumes. *Environmental Geosciences*, 21(2), 59–74. <https://doi.org/10.1306/eg.03101413009>
- Mishra, S., Schetter, J., Datta-Gupta, A., & Bromhal, G. (2021, March 1). Robust Data Driven Machine-Learning Models for Subsurface Applications: Are We There Yet? *Journal of Petroleum Technology*. <https://jpt.spe.org/robust-data-driven-machine-learning-models-for-subsurface-applications-are-we-there-ye>
- Mohaghegh, S. D. (2018). *Data-Driven Analytics for the Geological Storage of CO₂*. <https://www.routledge.com/Data-Driven-Analytics-for-the-Geological-Storage-of-CO2/Mohaghegh/p/book/978036773438>
- Quinlan, R. (1986). Induction of decision trees. 1 pp. 81e106). <https://hunch.net/wcoms-4771/quinlan.pdf>.
- Schuetter, J., Mishra, S. (2018). A data-analytics tutorial: Building predictive models for oil production in an unconventional shale reservoir. *SPE Journal*, 23(4), 1075–1089. <https://doi.org/10.2118/189969-pa>
- Scikit-learn. (n.d.). Retrieved July 17, 2020, from <https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart>. Stats models. (2020). <https://pypi.org/project/statsmodels/>.
- Stevens, Paul August 2012. *Towards an Ecosociology* - Paul Stevens, 2012 - SAGE Journals
- Stevens, P. (2012), *The Arab Uprisings and the International Oil Markets*, Chatham House Briefing Paper, February.
- Zhong, M., Schuetter, J., Mishra, S., & LaFollette, R. F. (2015). Do data mining methods matter? A Wolfcamp “Shale” case study. Society of Petroleum Engineers - SPE Hydraulic Fracturing Technology Conference 2015, 136–147. <https://doi.org/10.2118/173334-ms>

APPENDICES

Appendix A

Import Relevant packages

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style("darkgrid")
df = pd.read_excel('SPE_shale.xlsx')
```

```
In [2]: df.head().transpose()
```

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 23 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Well Number                           50 non-null    int64
1   Initial Pressure Estimate (psi)       50 non-null    int64
2   Reservoir Temperature (deg F)        50 non-null    int64
3   Net Pay (ft)                          50 non-null    int64
4   Porosity                              50 non-null    float64
5   Water Saturation                      50 non-null    float64
6   Oil Saturation                        50 non-null    float64
7   Gas Saturation                        50 non-null    float64
8   Gas Specific Gravity                  50 non-null    float64
9   CO2                                    50 non-null    float64
10  N2                                     50 non-null    float64
11  TVD (ft)                              50 non-null    float64
12  Spacing                               50 non-null    int64
13  # Stages                              50 non-null    int64
14  # Clusters                            50 non-null    int64
15  # Clusters per Stage                  50 non-null    float64
16  # of Total Proppant (MM Lbs)          50 non-null    float64
17  Lateral Length (ft)                  50 non-null    int64
18  Top Perf (ft)                        50 non-null    int64
19  Bottom Perf (ft)                     50 non-null    int64
20  Sandface Temp (deg F)                 50 non-null    float64
21  Static Wellhead Temp (deg F)         50 non-null    float64
22  Cumulative Gas Produced after 1 year, MCF 50 non-null    float64
dtypes: float64(13), int64(10)
memory usage: 9.1 KB
```

```
In [5]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df = pd.read_excel('SPE_shale.xlsx')
df.describe().transpose()
```

```
In [6]: plt.figure(figsize=(12,10))
sns.heatmap(df.corr(), annot=True, linecolor='white',
linewidth=2, cmap='coolwarm')
```

```
Out[6]: <AxesSubplot: >
```


Appendix B

Decision Tree Classifier Building in Scikit-learn

LET USE 70/30 FOR TRAINING AND TESTING

Let's define the x and y variables as follows:

```
In [7]: x=df.drop(['Cumulative Gas Produced after 1 year, MCF'], axis=1)
y=df['Cumulative Gas Produced after 1 year, MCF']
```

```
In [8]: # next import the train_test_split library and use a 70/30 split as follows:
from sklearn.model_selection import train_test_split
seed=1000
np.random.seed(seed)
X_train,X_test,y_train, y_test=train_test_split\
(x, y, test_size=0.30)
```

Next, import "DecisionTreeRegressor" from sklearn.tree as follows. For classification problems, simply use "DecisionTreeClassifier."

```
In [9]: from sklearn.tree import DecisionTreeRegressor
```

Let's define the decision tree

parameters as shown below:

```
In [10]: np.random.seed(seed)
dtree=DecisionTreeRegressor(criterion='squared_error', splitter='best',
max_depth=None, min_samples_split=4, min_samples_leaf=2,
max_features=None, ccp_alpha=0)
```

Let's apply "dtree" to "(X_train,y_train)" as follows:

Now that the model has been fit to training inputs and output, let's apply to predict "X_train" and "X_test" as follows. The main reason to apply to "X_train" is to be able to obtain the training accuracy on the model as well as the testing accuracy.

```
In [11]: dtree.fit(X_train,y_train)
y_pred_train=dtree.predict(X_train)
y_pred_test=dtree.predict(X_test)
```

Next, let's obtain the training and testing R2 as follows:

```
In [12]: corr_train=np.corrcoef(y_train, y_pred_train) [0,1]
print('Training Data R^2=',round(corr_train**2,4),'R=',
round(corr_train,4))
```

Training Data R²= 0.9618 R= 0.9807

```
In [13]: corr_test=np.corrcoef(y_test, y_pred_test) [0,1]
print('Testing Data R^2=',round(corr_test**2,4),'R=',
round(corr_test,4))
```

Testing Data R²= 0.7066 R= 0.8406

Optimized. Next, let's visualize the training actual versus prediction and testing

actual versus prediction as follows:

Optimized. Next, let's visualize the training actual versus prediction and testing

actual versus prediction as follows:

```
In [14]: plt.figure(figsize=(8,6))
plt.plot(y_train, y_pred_train, 'r.')
plt.xlabel('Training Actual')
plt.ylabel('Training Prediction')
plt.title('Cumulative Gas Produced after 1 year, MCF Training Actual Vs. Prediction')
```

```
Out[14]: Text(0.5, 1.0, 'Cumulative Gas Produced after 1 year, MCF Training Actual Vs. Prediction')
```

```
In [15]: plt.figure(figsize=(8,6))
plt.plot(y_test, y_pred_test, 'r.')
plt.xlabel('Testing Actual')
plt.ylabel('Testing Prediction')
plt.title('Cumulative Gas Produced after 1 year, MCF Testing Actual Vs. Prediction')
```

```
Out[15]: Text(0.5, 1.0, 'Cumulative Gas Produced after 1 year, MCF Testing Actual Vs. Prediction')
```

To properly evaluate the model from all aspects, let's also add MAE, MSE, and RMSE as follows:

```
In [16]: from sklearn import metrics
print('MAE:', round(metrics.mean_absolute_error(y_test,
y_pred_test),5))
print('MSE:', round(metrics.mean_squared_error(y_test,
y_pred_test),5))
print('RMSE:', round(np.sqrt(metrics.mean_squared_error(y_test,
y_pred_test)),5))
```

```
MAE: 1510.52203
MSE: 6744531.57498
RMSE: 2597.0236
```

```
In [17]: dtree.feature_importances_
```

```
Out[17]: array([0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 4.7660507e-03,
0.00000000e+00, 0.00000000e+00, 9.25832181e-03, 0.00000000e+00,
1.18975428e-02, 4.72649111e-02, 6.74044558e-02, 3.42315063e-01,
0.00000000e+00, 2.45926040e-04, 0.00000000e+00, 5.14282517e-01,
2.56465827e-03, 0.00000000e+00])
```

```
In [18]: feature_names=df.columns[:-1]
plt.figure(figsize=(8,6))
plt.show()
```

<Figure size 800x600 with 0 Axes>

```
In [19]: feature_imp=pd.Series(dtree.feature_importances_,
index=feature_names).sort_values(ascending=False)
sns.barplot(x=feature_imp, y=feature_imp.index)
plt.xlabel('Feature Importance Score Using Decision Tree')
plt.ylabel('Features')
plt.title("Feature Importance Ranking")
```

```
Out[19]: Text(0.5, 1.0, 'Feature Importance Ranking')
```

Note that the train_test_split was done randomly with 70% of the data used

as training and 30% of the data used as testing. Let's also do a five-fold cross-validation to observe the resulting average R2 as follows:

```
In [20]: from sklearn.model_selection import cross_val_score
np.random.seed(seed)
scores_R2=cross_val_score(dtree, x, y,cv=5,scoring='r2')
print(" R2_Cross-validation scores: {}".format(scores_R2))
```

```
R2_Cross-validation scores: [-0.69549639 -0.48062385 -0.73620648 -0.19568775 0.30827655]
```

```
In [21]: print(" Average R2_Cross-validation scores: {}".
format(scores_R2.mean()))
```

```
Average R2_Cross-validation scores: -0.3599475848899655
```

LET USE 75/25 FOR TRAINING AND TESTING

```
In [22]: x=df.drop(['Cumulative Gas Produced after 1 year, MCF'], axis=1)
y=df['Cumulative Gas Produced after 1 year, MCF']
```

```
In [23]: # next import the train_test_split library and use a 75/25 split as follows:
from sklearn.model_selection import train_test_split
seed=1000
np.random.seed(seed)
X_train,X_test,y_train, y_test=train_test_split\
(X, y, test_size=0.25)
```

```
In [24]: from sklearn.tree import DecisionTreeRegressor
```

```
In [25]: np.random.seed(seed)
dtree=DecisionTreeRegressor(criterion='squared_error', splitter='best',
max_depth=None, min_samples_split=4, min_samples_leaf=2,
max_features=None, ccp_alpha=0)
```

```
In [26]: dtree.fit(X_train,y_train)
y_pred_train=dtree.predict(X_train)
y_pred_test=dtree.predict(X_test)
```

```
In [27]: corr_train=np.corrcoef(y_train, y_pred_train) [0,1]
print('Training Data R^2=',round(corr_train**2,4),'R=',
round(corr_train,4))
```

Training Data R²= 0.9553 R= 0.9774

```
In [28]: corr_test=np.corrcoef(y_test, y_pred_test) [0,1]
print('Testing Data R^2=',round(corr_test**2,4),'R=',
round(corr_test,4))
```

Testing Data R²= 0.568 R= 0.7537

```
In [29]: plt.figure(figsize=(8,6))
plt.plot(y_train, y_pred_train, 'g.')
plt.xlabel('Training Actual')
plt.ylabel('Training Prediction')
plt.title('Cumulative Gas Produced after 1 year, MCF Training Actual Vs. Prediction')
```

Out[29]: Text(0.5, 1.0, 'Cumulative Gas Produced after 1 year, MCF Training Actual Vs. Prediction')

```
In [30]: plt.figure(figsize=(8,6))
plt.plot(y_test, y_pred_test, 'g.')
plt.xlabel('Testing Actual')
plt.ylabel('Testing Prediction')
plt.title('Cumulative Gas Produced after 1 year, MCF Testing Actual Vs. Prediction')
```

Out[30]: Text(0.5, 1.0, 'Cumulative Gas Produced after 1 year, MCF Testing Actual Vs. Prediction')

```
In [31]: from sklearn import metrics
print('MAE:', round(metrics.mean_absolute_error(y_test,
y_pred_test),5))
print('MSE:', round(metrics.mean_squared_error(y_test,
y_pred_test),5))
print('RMSE:', round(np.sqrt(metrics.mean_squared_error(y_test,
y_pred_test)),5))
```

MAE: 1500.39035
MSE: 7115398.33428
RMSE: 2667.4704

```
In [32]: dtree.feature_importances_
```

```
Out[32]: array([2.07285826e-03, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
2.53127789e-04, 1.50455074e-03, 0.00000000e+00, 3.73038210e-03,
0.00000000e+00, 0.00000000e+00, 6.91126112e-03, 0.00000000e+00,
9.31110928e-03, 3.81791274e-02, 5.27512498e-02, 3.65750647e-01,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 5.19535687e-01,
0.00000000e+00, 0.00000000e+00])
```

```
In [33]: feature_names=df.columns[:-1]
plt.figure(figsize=(8,6))
```

Out[33]: <Figure size 800x600 with 0 Axes>
<Figure size 800x600 with 0 Axes>

```
In [34]: feature_imp=pd.Series(dtree.feature_importances_,
index=feature_names).sort_values(ascending=False)
sns.barplot(x=feature_imp, y=feature_imp.index)
plt.xlabel('Feature Importance Score Using Decision Tree')
plt.ylabel('Features')
plt.title("Feature Importance Ranking")
```

Out[34]: Text(0.5, 1.0, 'Feature Importance Ranking')

```
In [35]: from sklearn.model_selection import cross_val_score
np.random.seed(seed)
scores_R2=cross_val_score(dtrees, x, y,cv=5,scoring='r2')
print(" R2_Cross-validation scores: {}".format(scores_R2))

R2_Cross-validation scores: [-0.69549639 -0.48062385 -0.73620648 -0.19568775  0.30827655]
```

```
In [36]: print(" Average R2_Cross-validation scores: {}".format(scores_R2.mean()))

Average R2_Cross-validation scores: -0.3599475848899655
```

LET USE 80/20 FOR TRAINING AND TESTING

```
In [37]: x=df.drop(['Cumulative Gas Produced after 1 year, MCF'], axis=1)
y=df['Cumulative Gas Produced after 1 year, MCF']
```

```
In [38]: # next import the train_test_split library and use a 85/25 split as follows:
from sklearn.model_selection import train_test_split
seed=1000
np.random.seed(seed)
X_train,X_test,y_train, y_test=train_test_split\
(X, y, test_size=0.20)
```

```
In [39]: from sklearn.tree import DecisionTreeRegressor
```

```
In [40]: np.random.seed(seed)
dtree=DecisionTreeRegressor(criterion='squared_error', splitter='best',
max_depth=None, min_samples_split=4, min_samples_leaf=2,
max_features=None, ccp_alpha=0)
```

```
In [41]: dtree.fit(X_train,y_train)
y_pred_train=dtree.predict(X_train)
y_pred_test=dtree.predict(X_test)
```

```
In [42]: corr_train=np.corrcoef(y_train, y_pred_train) [0,1]
print('Training Data R^2=',round(corr_train**2,4),'R=',
round(corr_train,4))
```

Training Data R²= 0.9578 R= 0.9787

```
In [43]: corr_test=np.corrcoef(y_test, y_pred_test) [0,1]
print('Testing Data R^2=',round(corr_test**2,4),'R=',
round(corr_test,4))
```

Testing Data R²= 0.5779 R= 0.7602

```
In [44]: plt.figure(figsize=(8,6))
plt.plot(y_train, y_pred_train, 'y.')
plt.xlabel('Training Actual')
plt.ylabel('Training Prediction')
plt.title('Cumulative Gas Produced after 1 year, MCF Training Actual Vs. Prediction')
```

```
Out[44]: Text(0.5, 1.0, 'Cumulative Gas Produced after 1 year, MCF Training Actual Vs. Prediction')
```

```
In [46]: from sklearn import metrics
print('MAE:', round(metrics.mean_absolute_error(y_test,
y_pred_test),5))
print('MSE:', round(metrics.mean_squared_error(y_test,
y_pred_test),5))
print('RMSE:', round(np.sqrt(metrics.mean_squared_error(y_test,
y_pred_test)),5))
```

```
MAE: 1899.08549
MSE: 9235036.62798
RMSE: 3038.92031
```

```
In [47]: dtree.feature_importances_
```

```
Out[47]: array([1.96209578e-03, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
1.43105535e-03, 5.69053198e-03, 0.00000000e+00, 8.28771883e-05,
0.00000000e+00, 0.00000000e+00, 6.54196020e-03, 0.00000000e+00,
1.19503413e-02, 3.61390385e-02, 4.55157071e-02, 3.61656372e-01,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 5.29030020e-01,
0.00000000e+00, 0.00000000e+00])
```

```
In [48]: feature_names=df.columns[:-1]
plt.figure(figsize=(8,6))
```

```
Out[48]: <Figure size 800x600 with 0 Axes>
<Figure size 800x600 with 0 Axes>
```

```
In [49]: feature_imp=pd.Series(dtree.feature_importances_,
index=feature_names).sort_values(ascending=False)
sns.barplot(x=feature_imp, y=feature_imp.index)
plt.xlabel('Feature Importance Score Using Decision Tree')
plt.ylabel('Features')
plt.title("Feature Importance Ranking")
```

```
Out[49]: Text(0.5, 1.0, 'Feature Importance Ranking')
```

```
In [50]: from sklearn.model_selection import cross_val_score
np.random.seed(seed)
scores_R2=cross_val_score(dtree, x, y, cv=5, scoring='r2')
print(" R2_Cross-validation scores: {}". format(scores_R2))
```

```
R2_Cross-validation scores: [-0.69549639 -0.48062385 -0.73620648 -0.19568775  0.30827655]
```

```
In [51]: print(" Average R2_Cross-validation scores: {}".
format(scores_R2.mean()))
```

```
Average R2_Cross-validation scores: -0.3599475848899655
```

Appendix C

Random Forest implementation using scikit-learn

LET USE 70/30 FOR TRAINING AND TESTING

```
In [52]: from sklearn.ensemble import RandomForestRegressor
```

```
In [53]: # next import the train_test_split library and use a 75/25 split as follows:
from sklearn.model_selection import train_test_split
seed=1000
np.random.seed(seed)
X_train,X_test,y_train, y_test=train_test_split\
(X, y, test_size=0.30)
```

```
In [54]: np.random.seed(seed)
rf=RandomForestRegressor(n_estimators=5000,
criterion='squared_error',max_depth=None, min_samples_split=4,
min_samples_leaf=2, max_features='auto', bootstrap=True,
n_jobs=-1)
```

```
In [55]: rf.fit(X_train,y_train)
y_pred_train=rf.predict(X_train)
y_pred_test=rf.predict(X_test)
corr_train=np.corrcoef(y_train, y_pred_train) [0,1]
print('Training Data R^2=',round(corr_train**2,4),'R=',
round(corr_train,4))
```

C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

```
warn(
Training Data R^2= 0.9404 R= 0.9697
```

```
In [56]: corr_test=np.corrcoef(y_test, y_pred_test) [0,1]
print('Testing Data R^2=',round(corr_test**2,4),'R=',
round(corr_test,4))
```

```
Testing Data R^2= 0.8436 R= 0.9185
```

```
In [57]: plt.figure(figsize=(8,6))
plt.plot(y_train, y_pred_train, 'b.',label='Training wells')
plt.xlabel('Training Actual')
plt.ylabel('Training Prediction')
plt.title('Training Actual Vs. Prediction')
plt.legend(fontsize=10)
```

```
Out[57]: <matplotlib.legend.Legend at 0x24b9d2578e0>
```

```
In [58]: plt.figure(figsize=(8,6))
plt.plot(y_test, y_pred_test, 'b.',label='Testing Wells')
plt.xlabel('Testing Actual')
plt.ylabel('Testing Prediction')
plt.title('Testing Actual Vs. Prediction')
plt.legend()
```

```
Out[58]: <matplotlib.legend.Legend at 0x24b9d611e70>
```

```
In [59]: feature_names=df.columns[:-1]
plt.figure(figsize=(8,6))
feature_imp=pd.Series(rf.feature_importances_,
index=feature_names).sort_values(ascending=False)
sns.barplot(x=feature_imp, y=feature_imp.index)
plt.xlabel('Feature Importance Score Using Random Forest')
plt.ylabel('Features')
plt.title("Feature Importance Ranking")
```

```
Out[59]: Text(0.5, 1.0, 'Feature Importance Ranking')
```

LET USE 75/25 FOR TRAINING AND TESTING

```
In [62]: from sklearn.ensemble import RandomForestRegressor
```

```
In [63]: # next import the train_test_split library and use a 75/25 split as follows:
from sklearn.model_selection import train_test_split
seed=1000
np.random.seed(seed)
X_train,X_test,y_train, y_test=train_test_split\
(x, y, test_size=0.25)
```

```
In [64]: np.random.seed(seed)
rf=RandomForestRegressor(n_estimators=5000,
criterion='squared_error',max_depth=None, min_samples_split=4,
min_samples_leaf=2, max_features='auto', bootstrap=True,
n_jobs=-1)
```

```
In [65]: rf.fit(X_train,y_train)
y_pred_train=rf.predict(X_train)
y_pred_test=rf.predict(X_test)
corr_train=np.corrcoef(y_train, y_pred_train) [0,1]
print('Training Data R^2=',round(corr_train**2,4),'R=',
round(corr_train,4))
```

```
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(
```

```
Training Data R^2= 0.9525 R= 0.976
```

```
In [66]: corr_test=np.corrcoef(y_test, y_pred_test) [0,1]
print('Testing Data R^2=',round(corr_test**2,4),'R=',
round(corr_test,4))
```

```
Testing Data R^2= 0.8737 R= 0.9347
```

```
In [67]: plt.figure(figsize=(8,6))
plt.plot(y_train, y_pred_train, 'y.',label='Trainig Wells')
plt.xlabel('Training Actual')
plt.ylabel('Training Prediction')
plt.title('Training Actual Vs. Prediction')
plt.legend()
```

```
Out[67]: <matplotlib.legend.Legend at 0x24b9f075d20>
```

```
In [60]: from sklearn.model_selection import cross_val_score
np.random.seed(seed)
scores_R2=cross_val_score(rf, x, y,cv=5,scoring='r2')
print(" R2_Cross-validation scores: {}".format(scores_R2))
```

```
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(
```

```
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(
```

```
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(
```

```
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(
```

```
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(
```

```
R2_Cross-validation scores: [-2.65359201 0.22012819 -0.32206354 -0.61958175 0.79659003]
```

```
In [61]: print(" Average R2_Cross-validation scores: {}".
format(scores_R2.mean()))
```

```
Average R2_Cross-validation scores: -0.5157038153259419
```

```
In [68]: plt.figure(figsize=(8,6))
plt.plot(y_test, y_pred_test, 'y .',label='Testing Wells')
plt.xlabel('Testing Actual')
plt.ylabel('Testing Prediction')
plt.title('Testing Actual Vs. Prediction')
plt.legend()
```

```
Out[68]: <matplotlib.legend.Legend at 0x24b9f077010>
```

```
In [69]: feature_names=df.columns[:-1]
plt.figure(figsize=(8,6))
feature_imp=pd.Series(rf.feature_importances_,
index=feature_names).sort_values(ascending=False)
sns.barplot(x=feature_imp, y=feature_imp.index)
plt.xlabel('Feature Importance Score Using Random Forest')
plt.ylabel('Features')
plt.title("Feature Importance Ranking")
```

Out[69]: Text(0.5, 1.0, 'Feature Importance Ranking')

```
In [70]: from sklearn.model_selection import cross_val_score
np.random.seed(seed)
scores_R2=cross_val_score(rf, x, y,cv=5,scoring='r2')
print(" R2_Cross-validation scores: {}".format(scores_R2))
```

C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

warn(
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

warn(
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

warn(
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

warn(
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

warn(
R2_Cross-validation scores: [-2.65359201 0.22012819 -0.32206354 -0.61958175 0.79659003]

```
In [71]: print(" Average R2_Cross-validation scores: {}".
format(scores_R2.mean()))
```

Average R2_Cross-validation scores: -0.5157038153259419

LET USE 80/20 FOR TRAINING AND TESTING

```
In [72]: from sklearn.ensemble import RandomForestRegressor
```

```
In [73]: # next import the train_test_split library and use a 75/25 split as follows:
from sklearn.model_selection import train_test_split
seed=1000
np.random.seed(seed)
X_train,X_test,y_train, y_test=train_test_split\
(X, y, test_size=0.20)
```

```
In [74]: np.random.seed(seed)
rf=RandomForestRegressor(n_estimators=5000,
criterion='squared_error',max_depth=None, min_samples_split=4,
min_samples_leaf=2, max_features='auto', bootstrap=True,
n_jobs=-1)
```

```
In [75]: rf.fit(X_train,y_train)
y_pred_train=rf.predict(X_train)
y_pred_test=rf.predict(X_test)
corr_train=np.corrcoef(y_train, y_pred_train) [0,1]
print('Training Data R^2=',round(corr_train**2,4),'R=',
round(corr_train,4))

C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(

Training Data R^2= 0.9536 R= 0.9765
```

```
In [76]: corr_test=np.corrcoef(y_test, y_pred_test) [0,1]
print('Testing Data R^2=',round(corr_test**2,4),'R=',
round(corr_test,4))

Testing Data R^2= 0.8777 R= 0.9369
```

```
In [77]: plt.figure(figsize=(8,6))
plt.plot(y_train, y_pred_train, 'g.',label= 'Training wells')
plt.xlabel('Training Actual')
plt.ylabel('Training Prediction')
plt.title('Training Actual Vs. Prediction')
plt.legend(fontsize=10)
```

```
Out[77]: <matplotlib.legend.Legend at 0x24b9d2fead0>
```

```
In [78]: plt.figure(figsize=(8,6))
plt.plot(y_test, y_pred_test, 'g.',label='Testing Wells')
plt.xlabel('Testing Actual')
plt.ylabel('Testing Prediction')
plt.title('Testing Actual Vs. Prediction')
plt.legend()
```

```
Out[78]: <matplotlib.legend.Legend at 0x24ba02a0d90>
```

```
In [79]: feature_names=df.columns[:-1]
plt.figure(figsize=(8,6))
feature_imp=pd.Series(rf.feature_importances_,
index=feature_names).sort_values(ascending=False)
sns.barplot(x=feature_imp, y=feature_imp.index)
plt.xlabel('Feature Importance Score Using Random Forest')
plt.ylabel('Features')
plt.title("Feature Importance Ranking")
```

```
Out[79]: Text(0.5, 1.0, 'Feature Importance Ranking')
```

```
In [80]: from sklearn.model_selection import cross_val_score
np.random.seed(seed)
scores_R2=cross_val_score(rf, x, y,cv=5,scoring='r2')
print(" R2_Cross-validation scores: {}". format(scores_R2))
```

```
C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(

C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(

C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(

C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
warn(

R2_Cross-validation scores: [-2.65359201  0.22012819 -0.32206354 -0.61958175  0.79659003]
```

```
In [81]: print(" Average R2_Cross-validation scores: {}".
format(scores_R2.mean()))

Average R2_Cross-validation scores: -0.5157038153259419
```

Appendix D

Variables of Multivariable Correlation Plot for Reservoir Parameters

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df = pd.read_excel('SPE_1yr_Gas_Draw_a.xlsx')
df.describe().transpose()
```

Out[2]:

	count	mean	std	min	25%	50%	75%	max
Initial Pressure Estimate (psi)	50.0	6313.780000	2963.478970	2200.0000	4300.000000	5164.000000	9929.250000	12223.00000
Reservoir Temperature (deg F)	50.0	211.160000	93.430398	115.0000	134.000000	144.500000	323.000000	379.00000
Net Pay (ft)	50.0	163.380000	56.790661	56.0000	136.000000	164.500000	208.750000	268.00000
Porosity	50.0	0.070614	0.013419	0.0500	0.060300	0.068450	0.083900	0.10000
Water Saturation	50.0	0.301070	0.087444	0.1838	0.210000	0.310050	0.362500	0.47000
Oil Saturation	50.0	0.107326	0.248963	0.0000	0.000000	0.000000	0.000000	0.74000
Gas Saturation	50.0	0.591604	0.272580	0.0000	0.574275	0.676550	0.790000	0.81620
Gas Specific Gravity	50.0	0.612286	0.095459	0.5700	0.570000	0.570000	0.594750	0.95130
CO2	50.0	0.011626	0.017981	0.0000	0.000000	0.000000	0.024575	0.05800
N2	50.0	0.000304	0.000770	0.0000	0.000000	0.000000	0.000400	0.00450
Cumulative Gas Produced after 1 year, MCF	50.0	4378.222749	3273.300086	25.1260	1618.665060	3792.715095	6355.904765	13094.84705

Appendix E

Variables of Multivariable Correlation Plot for Operational Parameters

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df = pd.read_excel('SPE_1yr_Gas_Draw_b.xlsx')
df.describe().transpose()
```

Out[1]:

	count	mean	std	min	25%	50%	75%	max
TVD (ft)	50.0	9020.337103	2160.285784	5707.639	7389.30875	7794.690000	11755.378798	12668.00000
Spacing	50.0	1220.200000	297.468846	700.000	1000.00000	1200.000000	1500.000000	1850.00000
# Stages	50.0	45.180000	20.075053	7.000	30.25000	47.000000	62.500000	89.00000
# Clusters	50.0	276.460000	160.971820	49.000	141.00000	234.000000	444.000000	735.00000
# Clusters per Stage	50.0	6.078200	2.160534	3.000	5.00000	5.000000	7.000000	15.00000
Lateral Length (ft)	50.0	7867.840000	2354.971826	2268.000	5990.00000	7480.000000	9800.000000	13011.00000
Top Perf (ft)	50.0	9204.480000	2224.347208	5900.000	7548.50000	8133.000000	12082.000000	13153.00000
Bottom Perf (ft)	50.0	17054.920000	3608.288191	10049.000	14360.75000	16192.000000	20089.500000	23203.00000
Cumulative Gas Produced after 1 year, MCF	50.0	4378.222749	3273.300086	25.126	1618.66506	3792.715095	6355.904765	13094.84705

Appendix F

Similarity Report

Assignments

Students

Grade Book

Libraries

Calendar

Discussion

Preferences

[Cari Alair](#) | [User Info](#) | [Messages](#) | [Instructor](#) | [English](#) | [Community](#) | [Help](#) | [Logout](#)

NOW VIEWING: HOME > MASTER > JOHN KANINDA MFUTA

About this page

This is your assignment inbox. To view a paper, select the paper's title. To view a Similarity Report, select the paper's Similarity Report icon in the similarity column. A ghosted icon indicates that the Similarity Report has not yet been generated.

JOHN KANINDA MFUTA

INBOX | NOW VIEWING: NEW PAPERS ▾

Submit File
Online Grading Report | Edit assignment settings | Email non-submitters

	AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE
<input type="checkbox"/>	John Kaninda Mluta	ABSTRACT	0% ■	--	--	<input type="checkbox"/>	2129573450	11-Jul-2023
<input type="checkbox"/>	John Kaninda Mluta	CHAPTER 1	11% ■	--	--	<input type="checkbox"/>	2129573258	11-Jul-2023
<input type="checkbox"/>	John Kaninda Mluta	CHAPTER 2	7% ■	--	--	<input type="checkbox"/>	2129572945	11-Jul-2023
<input type="checkbox"/>	John Kaninda Mluta	CHAPTER 3	2% ■	--	--	<input type="checkbox"/>	2129573264	11-Jul-2023
<input type="checkbox"/>	John Kaninda Mluta	CHAPTER 4	12% ■	--	--	<input type="checkbox"/>	2129573445	11-Jul-2023
<input type="checkbox"/>	John Kaninda Mluta	CONCLUSION	0% ■	--	--	<input type="checkbox"/>	2129573458	11-Jul-2023
<input type="checkbox"/>	John Kaninda Mluta	THESIS	12% ■	--	--	<input type="checkbox"/>	2129573816	11-Jul-2023

Copyright © 1998 – 2023 Turnitin, LLC. All rights reserved.

[Privacy Policy](#) | [Privacy Pledge](#) | [Terms of Service](#) | [EU Data Protection Compliance](#) | [Copyright Protection](#) | [Legal FAQs](#) | [Helpdesk](#) | [Research Resources](#)

Appendix G
Ethical Approval Letter



YAKIN DOĞU ÜNİVERSİTESİ
ETHICAL APPROVAL DOCUMENT

Date: 20/06/2023

To the Institute of Graduate Studies

The research project titled “**Analysis and Performance Prediction of Shale Wells Using Data Analytics and Machine Learning**” has been evaluated. Since the researcher will not collect primary data from humans, animals, plants or earth, this project does not need through the ethics committee.

Title: Prof. Dr.

Name Surname: Cavit ATALAR

Signature:

Role in the Research Project: Supervisor