2021-11

Production Forecasting for the Duvernay Shale: Comparing Analytical and Machine Learning Methods

Malaieri, Mohammadreza

UNIVERSITY OF CALGARY

Production Forecasting for the Duvernay Shale:

Comparing Analytical and Machine Learning Methods

by

Mohammadreza Malaieri

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF ENGINEERING

GRADUATE PROGRAM IN CHEMICAL AND PETROLEUM ENGINEERING

CALGARY, ALBERTA

NOVEMBER, 2021

Abstract

With the advance in drilling horizontal wells and hydraulic fracturing, shale production has increased dramatically in the past two decades. Shale formations are characterized by natural fractures, low-permeability rocks, and nanopores associated with organic matter, clay minerals, carbonate, and various silts. In shale, gas content is mainly composed of free gas in the nanopores, adsorbed gas in the rocks, and other components such as carbon dioxide, nitrogen, and hydrogen sulfide. The co-existence of these components with different sizes and properties has posed a significant challenge for investigating shale. Due to complex transport mechanisms and the existence of natural and hydraulic fracture networks, building reasonable reservoir simulation models and production forecasting is not entirely feasible with the existing simulators. Still, it is a challenging task for petroleum engineers. While simulation methods, including analytical models, semi-analytical models, and numerical simulation, could lead to reasonable production forecast, they are expensive and time-consuming and require many data such as well test data and well logs which are not always accessible.

In contrast, Decline Curve Analysis (DCA) has the advantages of simplicity and efficiency for hydrocarbon production forecasting and mainly requires production data. In the past years, different DCA models have been proposed and benchmarked to meet industrial needs. DCA models have shown to be efficient for conventional reservoirs.

When applied to shale wells, DCA has many shortcomings. Several authors have tried to overcome some of the well-known shortcomings of DCA. Nevertheless, many facts remain that make the use of DCA suboptimal. One of the main criticisms of DCA is its lack of sensitivity to major physical phenomena in shale wells that have to do with the fluid flow, the hydraulic fracture, and the

reservoir characteristics. In cases like Duvernay shale reservoirs where short production periods are available, the use of DCA becomes increasingly problematic.

In this study, different methods of production forecasting, including reservoir simulation, decline curve analysis, machine learning, and time series will be reviewed, and their limitations will be discussed. Then these methods have been used to predict and forecast gas production in Duvernay play in different scales of wells and fields.

Acknowledgments

I would like to thank Dr. Roman Shor for being my advisors on this thesis. He provided excellent academic and technical advice throughout the research process. I would also thank Dr. Roberto Aguilera and Dr. Matthew Clarke for being my examiner and their comments and advice to improve the work.

I would also like to thank IHS Markit for accessing their repositories through the University of Calgary gateway. I would also appreciate TIBCO for accessibility to use the Spotfire application.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations, & Nomenclature

a	Constants (3-17), (3-21)
$a(.)$	Activation function
ACF	Autocorrelation Function
ADF	Augmented Dickey-Fuller
AIC	Akaike's Information Criterion
AICc	Corrected AIC
ANN	Artificial Neural Network
AR(p)	Autoregressive model of order p
ARCH	Auto-Regressive Conditional Heteroscedastic model
ARIMA	Auto-Regressive Integrated Moving Average
b	Constant (3-5)
b	Bias (4-5)
BDF	Boundary-dominated flows
BIC	Bayesian Information Criterion
c_t	Cells state (memory) at timestamp t
$CV(\hat{f})$	Cross-validated prediction error
D_1, D_2	Constant (3-7)
D_∞	Decline constant at the infinite time
DCA	Decline Curve Analysis
D_i	Decline rate
EUR	Estimated Ultimate Recovery
G_p	Cumulative gas production
$h(x)$	Hypotheses
$J(\theta)$	Cost function
k	Carrying capacity (3-21)
k	Number of folds (5-14)
L	Likelihood (7-4)
LGM	Logistic Growth Model
m	Constants (3-17)
m	Number of sample points (4-1)
MA(q)	Moving average model of order q
MAE	Mean Average Error
MEA	Mean Absolute Error
MHD	Modified Hyperbolic Decline Model
ML	Machine Learning
MSE	Mean square error
n	Time exponent (3-11)
n	Hyperbolic exponent (3-21)
n	Number of features (4-1)
NN	Neural Network
NNAR	Neural Network Auto Regression
N_p	Cumulative production rate
PACF	Partial Autocorrelation Function
PCA	Principal Component Analysis

PLE	Power Law Exponential Decline Model
q	Flow rate
q_i	Initial rate
r_{21}, r_{21}	Cumulative production ratio for two and three years (3-13)
R^2	Correlation coeficient
RF	Random Forest
RMSE	Root Mean Squared Error
R_t	Remainder component
RTA	Rate Transient Analysis
$S.E(\hat{\rho})$	Standard error
SED	Stretched Exponential Decline Model
SPM	Shot per Meter
S_t	Seasonal component
SVM	Support Vector Machine
t	Time
T	Total periods in time series (7-5)
TVD	True Verical Depth
TD	Total Depth
T_t	Trend-cycle component
w	Weight
$W_{(i,f,o)}$	Weight for the respective gate neurons
x	Input variable/feature
y	Output variable/feature
$\hat{y}$	Predicted value
y_t	Predictors/terms
y'_t	Differenced series
z	Outcome (4-5)
α	Training rate (4-3)
α, β, N	Constant (3-6)
βl, βe	Constant (3-23)
$\Gamma(s,x)$	Incomplete gamma function
$\Gamma(t)$	Gamma function
ε_t	White noise
θ_i	Weights
θ^T	Transposed weight vector
$\hat{\rho}$	Coefficient estimation
ρ_k	Correlation between all pairs of data points with k steps apart
τ	The characteristic time for stretched exponential
φ_i	Timeseries parameters
WSM	World Stress Map

1 INTRODUCTION

Shale production is a revolutionized landscape in the oil and gas industry and has been noted as a game-changer energy in the twenty-first century. The scientific and engineering knowledge about the completion, stimulation, and interaction between induced and natural fracture during production from shale is not vividly described. This shortcoming makes the quick decision in day-to-day operation tedious and even impossible. A complete understanding of the physical mechanism of fluid flow and transport phenomena is a preliminary requirement for shale development.

Production forecasting which is a preliminary step for future field development and strategic decision, directly inferred and estimated from a time consuming and expensive reservoir simulation process, and needs a team of geoscientists and engineer to build and QC the reservoir model and require many expensive data which is not always available in the field. Traditional analytical approaches lack enough accuracy to answer all the questions required by engineers. Even with advancement and modification to adopt them with shale reserve, there are still considered alternative and quick methods of production estimation. Operators collect a significant amount of data from their operation that contains the information and knowledge we need. However, analyzing this vast amount of data is not possible by human cognition, and with the recent advancement in computer science and consequently in data science, now we can analyze a large amount of data in a short time, which previously was impossible.

Artificial intelligence (AI) and Machine Learning (ML) are generated due to this scientific revolution and can mimic the reservoir behavior by implementing fast learning algorithms and then be quickly responsive to the new conditions.

Different ML algorithms will be examined and tested in this study to identify an appropriate module for production forecasting in Duvernay shale. For this purpose, various databases from geological to engineering data have been collected and analyzed.

Chapter 2 briefly explains the steps of building a dynamic reservoir model based on the conventional approach of reservoir simulation. The pros and cons of this method will be discussed.

Chapter 3 presents different analytical approaches to production forecasting. This chapter will discuss the origin, derivatives, pros, and cons of each method. Starting with traditional methods of Decline Curve Analysis (DCA) and then the most recent attempts conducted to adopt this method with the shale situation will be discussed.

Chapter 4 explains the most common ML algorithms, including regression, neural network, support vector machine, random forest, and boosting. Then the ML workflow that applied in this research is illustrated and explained. A different method of ML model testing and validation is discussed.

Chapter 5 specified for time series approach, and different time-based algorithms have been extensively explained.

Chapter 6 is the starting point of implemented procedure for production forecasting in Duvernay shale. Collected datasets have been analyzed statistically and visualized to describe the pattern and trend of different parameters. The relationship between these operational parameters with production has been graphically illustrated and discussed.

Chapter 7 presents the steps required to build different models based on DCA, ML, and time series analysis (TSA). Different packages from R and Python have been used to build, validate and visualize the models.

Chapter 8 concludes the current research results and then suggests the potential future works that other researchers can conduct.

The main goal of conducting this research is to build and compare different methods of production forecasting (analytical and data-driven) for Duvernay shale and evaluate their advantages and disadvantages. This research suggests that the time-series based approach is more reliable in shale plays due to these reasons:

1- Shale production is highly tied to the market and fluctuation of oil and gas prices
2- Seasonal changes can impact the production performance and needs to be reflected in the prediction model
3- Historical trend and seasonal cyclic can be continued or repeated in near future

2 PRODUCTION FORECASTING WITH RESERVOIR SIMULATION MODEL

2.1 Reservoir Simulation

In general, simulation is a theoretical or a physical representation of an operation and an imitation of its system/processes in real life. Reservoir simulation is a branch of reservoir engineering that combines physics, mathematics, and computer programming to build a reservoir model that allows the analysis and the prediction of the fluid behavior in the reservoir over time. It can be considered the process of mimicking fluid flow behavior in a petroleum reservoir system using either physical or mathematical models [1]. The use of reservoir simulation has grown during the past three decades because of its ability to predict the future performance of oil and gas reservoirs over a wide range of operating conditions [2]. Reservoir simulation mainly consists of:

1- A geological/structural grid to describe formations
2- A fluid flow model to describe fluid movement in porous media
3- A wellbore model to describe flow in and out of the reservoir

To perform a reservoir simulation job, it is necessary to conduct several complex studies usually made by teams of specialists from different disciplines (Figure 2-1) due to the large amount of data required to prepare the simulation input dataset. The main elements of a simulation study include:

1- matching field history

2- making a forecast based on the existing operational data
3- evaluating different operating scenarios

Numerical modeling like the finite difference is a major reservoir simulation task that covers three main physical concepts to describe fluid dynamics: conservation of mass, fluid phase behavior, and fluid flow through porous media (Darcy equation).

The simulation model computes the saturation change of three phases (oil, water, and gas) and the pressure of each phase in each grid block at each time step. As a result of declining pressure, gas will be liberated from the oil in a reservoir depletion study. If pressures increase due to water or gas injection, the gas is redissolved into the oil phase.

In a developed field, history matching usually requires to compare historical field production and pressures with the calculated values. The model's parameters are adjusted until a reasonable match is achieved on a field basis and usually for all wells. Commonly, producing water cuts or water-oil ratios and gas-oil ratios are matched.

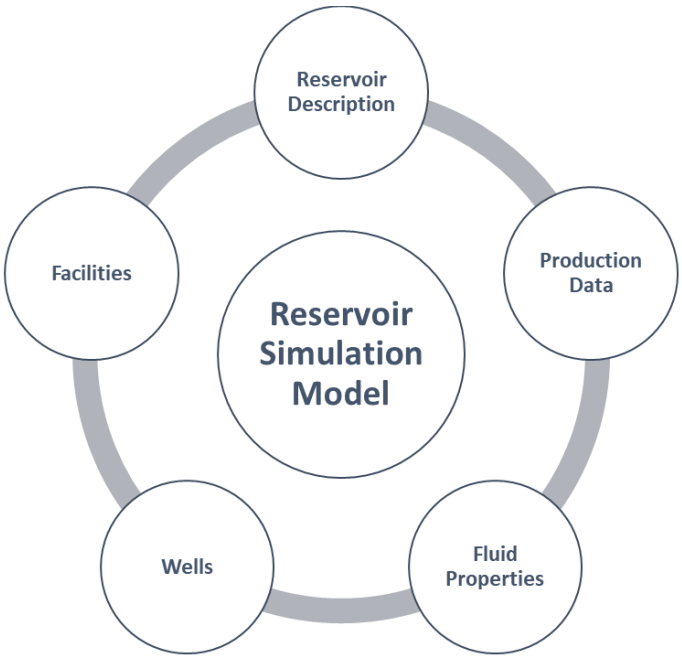


Figure 2-1: Reservoir simulation studies elements

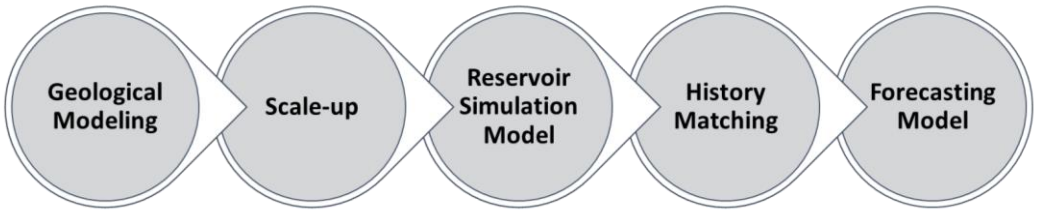


Figure 2-2: Reservoir simulation steps

2.2 Current Challenges in Reservoir Simulation

Reservoir simulation and numerical modeling have been around for a long time and are primary field development planning tools. Drilling new wells, shutting in some of the production wells, adding injection wells, and many other development plans are critical decisions that are usually made based on numerical simulation models.

Using a comprehensive numerical model, it is possible to study all possible production and injection scenarios. Unfortunately, to build a comprehensive numerical model, hundreds and sometimes thousands of simulation runs should be conducted, computationally expensive and time-consuming.

With the recent advance in developing smart fields, the need to process the data in real-time has become more prominent. Smart fields refer to intelligent technologies that collect data from sensors installed in wells and transforming them into valuable information to both control rooms and smart devices such as Inflow Control Devices (ICD) or Inflow Control Valves (ICV) [3]. This smart system should be able to monitor all subsets in real-time and be responsive to unexpected situations and required to make a quick decision; therefore, reducing the computational time of the simulation models is critical. The validity of sensor information and the completeness of data will determine the efficiency and accuracy of this technology.

The confidence level of a reservoir model is directly dependent on the input used to build the static model and also the quality of history match [4]. To overcome the computational expense, the

model is commonly upscaled to a finer model. Nevertheless, updating and maintenance a reservoir model takes a long time for history matching. On the other hand, these models require very sophisticated platforms to run. As the size of the reservoir models grows, the time required to run different scenarios also increases. Therefore, it is impractical to use a comprehensive simulation model on a routine basis.

The time spent to develop the reservoir simulation models, the cost of the platforms used, and the labor involved are the main parameters for calculating the return on investment in reservoir simulation.

The numerical models become even more complex and computationally intensive with developing unconventional fields like shale reserves and the use of advanced technology like hydraulic fracturing. Although some interesting work has been performed, especially in the area of transport at the micro-pore level, they have not yet found their way into the popular simulation models that are currently being used by the industry [5].

While most of the recent reservoir simulations and shale modeling can model the shale reservoir, they usually vary on how they handle the massive multi-cluster, multistage hydraulic fractures that are the main reason for economic oil and gas production from shale reservoirs [5]. Consequently, alternative methods and techniques have been developed and applied to solve the issue. One of the approaches to the issue was using traditional Proxy models to speed up the process and perform the necessary analyses; however, as proxy models are simplified versions of an existing complex numerical model, the results carry less accuracy and higher uncertainties [6].

3 PRODUCTION FORECASTING WITH DECLINE CURVES

3.1 Introduction

Hydrocarbon production forecasting includes the estimation of the ultimate recoveries (EUR) and the lifetimes of wells, which are vital factors for decision-making in the oil and gas industry as they can directly impact economic evaluation and field development planning.

Although math-based forecasting models (e.g., grid-based reservoir simulation models) have been developed over the past decades, decline curve analysis (DCA) is still widely used because of its simplicity. The mathematical formulations of DCA models are simple, with only a few parameters, and only production data are required to calibrate the parameters. Following, we will discuss more different DCA models.

3.2 Decline Curve Analysis (DCA)

Decline Curve Analysis (DCA) is undoubtedly the commonly used technique to estimate ultimate recovery (EUR) and production forecasting due to its simplicity and consistency. DCA consists of fitting a mathematically modeled curve with a plot of flow rate vs. time. In essence, DCA models are regressions for historical production data. The development of modem DCA was presented by Arps [7]. Arps developed a family of functional relations based on the hyperbolic decline model to analyze flow rate data. He provided a variety of exponential, hyperbolic, and harmonic rate

decline relations widely used in the industry as a benchmark for analysis and interpretation of production data.

3.2.1 Arps Decline Curve

The classical DCA model was proposed by Arps, 1945 [7]. In the Arps model, bottomhole pressure is fixed, the skin factor is constant, and the flow regime is boundary-dominated flow (refers to a late-time flow regime that starts when the drainage radius of the well reaches the reservoir boundaries and reservoir is in a state of pseudo-equilibrium [8]). To derive the DCA model, the concept of loss ratio was first introduced. The loss ratio is defined as the ratio between the production drop of the current time step and that of the previous time step.

He categorized the decline curve using the loss ratio method and then defined the rate/time and rate/cumulative production. He described three types of decline curve models: exponential, harmonic, and hyperbolic decline curves. The hyperbolic decline curve is a generic model which exponential and harmonic decline curves can be derived [9]. The decline curve has three parameters $\boldsymbol{q_i}$ (initial production), $\boldsymbol{D_i}$ (decline factor), and $\boldsymbol{b}$ (hyperbolic decline exponent). The following equation defines these three decline curve models:

$$D(t) = -\frac{1}{q(t)}\frac{dq(t)}{dt} \tag{3-1}$$

Loss ratio or decline parameter can be defined as:

$$\frac{1}{D(t)} = -\frac{q(t)}{dq(t)/dt} \tag{3-2}$$

Taking the derivative of the loss ratio gives:

$$b(t) = \frac{d}{dt}\left[\frac{1}{D(t)}\right] = -\frac{d}{dt}\left[\frac{q(t)}{\frac{dq(t)}{dt}}\right] \tag{3-3}$$

If we consider the decline parameter constant, then the solution of the ordinary differential equation yields the exponential decline, which can be derived for the case of pseudo-steady-state (or boundary-dominated) flow in a closed reservoir producing above bubble point pressure and being produced at a constant wellbore flowing pressure. The exponential relation is given as:

$$q(t) = q_i \exp[-D_i t] \tag{3-4}$$

A hyperbolic rate decline relation is obtained if the ordinary differential equation is solved for the rate function. The ***b*** value is constant in the hyperbolic relation and changes from 0 to 1:

$$q(t) = \frac{q_i}{(1 + bD_i t)^{1/b}} \tag{3-5}$$

The most important parameter in DCA that drives the shape of the decline curve is the **b** value. Table 3-1 shows the approximate range of **b** values for various reservoir drive mechanisms. As can be seen, unconventional shale and tight gas reservoirs typically have a **b** value over one due to the long transient period caused by low permeability. In conventional reservoirs with hyperbolic decline, **b** generally is between 0 and 1 depending on the reservoir drive mechanism. The Arps relations for flow rate and cumulative are summarized in Table 3-2 [10].

Table 3-1: Reservoir Drive Mechanism Versus b Values [10]

b Value	Reservoir Drive Mechanism
0	Single-phase liquid expansion (oil above bubble point)
0.1 – 0.4	Solution gas drive
0.4 – 0.5	Single-phase gas expansion
0.5	Effective edge water drive
0.5 – 1	Layered reservoirs
> 1	Transient (tight gas, shales)

Table 3-2: Arps relations for flow rate and cumulative production [10]

Case	Arps Flow Rate Relations	Arps Cumulative Production Relations
Exponential: (b=0)	$q(t) = q_i \exp(-D_i t)$	$N_p(t) = \frac{1}{D_i}[q_i - q(t)]$
Hyperbolic: (0<b<1)	$q(t) = \frac{q_i}{[1+bD_i t]^{1/b}}$	$N_p(t) = \frac{q_i}{(1-b)D_i}\left[1 - \left(\frac{q(t)}{q_i}\right)^{1-b}\right]$
Harmonic: (b=1)	$q(t) = \frac{q_i}{[1+D_i t]}$	$N_p(t) = \frac{q_i}{D_i}\ln\left(\frac{q_i}{q(t)}\right)$

After identifying a decline model, the model parameters can be determined by fitting the data to the selected model. The ***b*** value can be estimated for the exponential decline model based on the straight-line slope in ***log(q)*** versus ***t*** plot. The ***b*** value can also be determined based on the straight-line slope in the plot of ***q*** versus N_p. The harmonic decline model can estimate the ***b*** value based on the straight-line slope in the plot of ***log(q)*** versus ***log(t)***. it also can be estimated based on the slope of the straight line in the plot of N_p versus ***log(q)***. For the hyperbolic decline model, determining ***a*** and ***b*** values is somewhat tedious [11]. The procedure is shown in Figure 3-1.

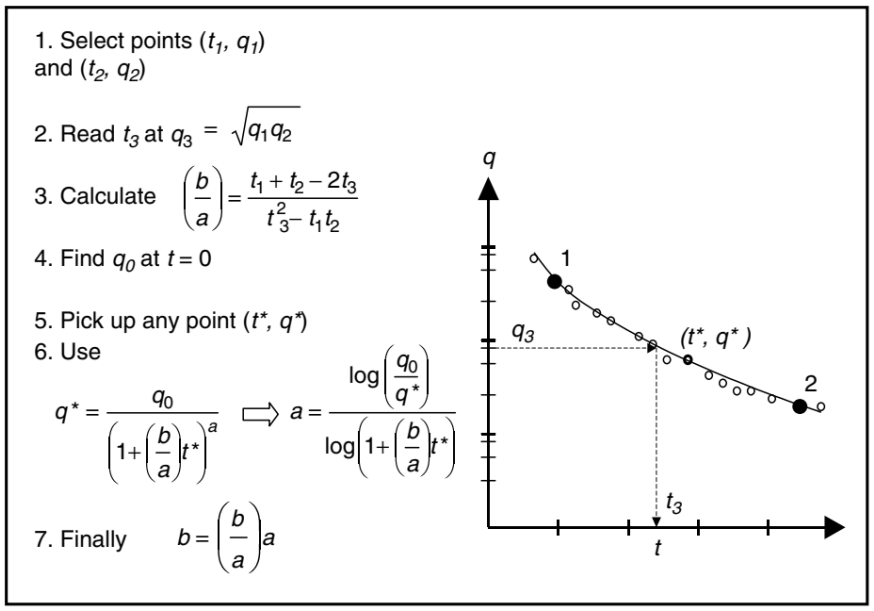


Figure 3-1: procedure for determining a (1/D) and b value [11]

It is noted that Arps decline curve relations can be considered as an industry standard and regularly used to forecast production in conventional reservoirs. However, it can also be stated that simplified time-rate (decline) models such as Arps decline curves cannot accurately capture all elements of the performance behavior in unconventional reservoirs. The Arps model often tends to overestimate the EUR for shale gas wells because it assumes that a boundary-dominated flow regime prevails [12, 13]. Since most shale gas wells rarely reach the boundary-dominated flow regime, the Arps model cannot be applied directly to shale gas reservoirs without significant

modifications. Various new models were proposed to model the boundary-dominated flow phase better and will be discussed in the following.

3.2.2 Modified Hyperbolic Decline Model

Arps model was initially designed for pseudo-radial boundary-dominated flows (BDF), which occur in medium–high-permeability formations. However, the fracture-dominated flow regimes dominate for tight-gas shale reservoirs, and the late-time-flow regime is rarely reached. Meanwhile, the Arps decline model leads to an unbounded and unrealistic estimate of EUR for $b \geq 1$ [14]. To solve this issue, Robertson [15] proposed the following Modified Hyperbolic Decline (MHD) model:

$$q(t) = q_i \frac{(1-\beta)^N e^{-at}}{(1-\beta e^{-at})^N} \tag{3-6}$$

where $\boldsymbol{a}, \boldsymbol{\beta}$ and $\boldsymbol{N}$ are all positive constant.

Seshadri and Mattar [16] proposed to shift hyperbolic decline in the early life of a well to exponential decline in the late-life by imposing a predetermined decline rate, $\boldsymbol{D}^*$. Once the decline rate reaches $\boldsymbol{D}^*$, the decline curve switches from a hyperbolic function to an exponential function. The equivalent MHD model is written as

$$q(t) = \begin{cases} q_i(q + bD_1 t)^{-1/b} & (D < D^*) \\ q_i e^{-D_2 t} & (D \geq D^*) \end{cases} \tag{3-7}$$

where $\boldsymbol{q_i}$ is the initial rate, in bbl/day, and $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ are decline rates.

3.2.3 Power Law Exponential Decline Model

Ilk et al. developed power law exponential (PLE) decline by modifying Arps exponential decline [17]. This aproach was developed particurlarly for tight gas wells to model the decline in a transient period of production data. PLE propose the below formula to obtain the $\boldsymbol{b}$ value, which is assumed to be constant in the Arps equation:

$$b = D_\infty + D_1 t^{-(1-n)} \tag{3-8}$$

Where $\boldsymbol{D_\infty}$, $\boldsymbol{D_1}$ are the decline constant at the infinite and initial time, and *n* is the time exponent. This equation shows that a decaying power function can approximate the loss ratio with constant behavior at late production time. The PLE decline model is defined as:

$$q = q_i \exp\left[-D_\infty t - \frac{D_1}{n} t^n\right] \tag{3-9}$$

The equation can be reduced to power-law loss ratio as defined below:

$$q = q_i \exp[-D_\infty t - D_i t^n] \tag{3-10}$$

$\boldsymbol{D_i}$ denotes the initial decline rate per year. In contrast to Arp method, the PLE method consider the $\boldsymbol{b}$ value to be as a declining function rather than a constant value. Moreover, the PLE model makes it easier to match the production data in transient and boundary-dominated regions without overestimating the reserve.

3.2.4 Stretched Exponential Decline Model

Valko [18] developed a newer form of the Arps model that inclueds a bounding component to provide a constrain on EUR. This empirical model, unlike other DCA models, has a physics basis and is governed by a defining differential equation:

$$\frac{dq}{dt} = -n\left(\frac{t}{\tau}\right)^n \frac{q}{t} \tag{3-11}$$

$\boldsymbol{\tau}$ denotes a characteristic time for stretched exponential, and $\boldsymbol{n}$ is the time exponent. The equation can then be separated and integrated to take on a rate form:

$$q(t) = q_i \exp\left[-\left(\frac{t}{\tau}\right)^n\right] \tag{3-12}$$

This method is comparable to the PLE model and can be equated to a particular instance where $\boldsymbol{D_\infty}$ is equal to zero and $\boldsymbol{\tau}$ is equal to $\boldsymbol{1/D_i}$; it does, however, ignores the behavior at late times. To determine $\boldsymbol{\tau}$, we need to solve these equations:

$$\frac{\Gamma\left[\frac{1}{n}\right]-\Gamma\left[\frac{1}{n},\left(\frac{24}{\tau}\right)^{n}\right]}{\Gamma\left[\frac{1}{n}\right]-\Gamma\left[\frac{1}{n},\left(\frac{12}{\tau}\right)^{n}\right]}=r_{21}\,,\frac{\Gamma\left[\frac{1}{n}\right]-\Gamma\left[\frac{1}{n},\left(\frac{36}{\tau}\right)^{n}\right]}{\Gamma\left[\frac{1}{n}\right]-\Gamma\left[\frac{1}{n},\left(\frac{12}{\tau}\right)^{n}\right]}=r_{31} \tag{3-13}$$

where $\boldsymbol{r_{21}}$ and r_{31} are the cumulative production ratio for two and three years, respectively. $\boldsymbol{\Gamma(t)}$ and $\boldsymbol{\Gamma(s,x)}$ are the gamma function and incomplete gamma function, respectively. They are defined as follows:

$$\Gamma(t)=\int_0^{\infty}x^{t-1}e^{-x}\,dx \tag{3-14}$$

$$\Gamma(s,x)=\int_x^{\infty}x^{s-1}e^{-t}\,dt \tag{3-15}$$

Stretched exponential outperforms PLE by providing the cumulative time relation as follows:

$$Q=\frac{q_i\tau}{n}\left\{\Gamma\left[\frac{1}{n},\left(\frac{t}{\tau}\right)^{n}\right]\right\} \tag{3-16}$$

However, to accurately estimate $\boldsymbol{\tau}$ and $\boldsymbol{n}$, the model requires a sufficiently long production time (usually more than 36 months) and solving the non-linear equation, which is complicated.

3.2.5 Duong Decline

Duong developed the rate decline analysis for fractured shale reservoirs [19]. The long-term linear flow was taken into account in this model. This model may be defined as:

$$q(t)=q_i t(a,m)+q_{\infty} \tag{3-17}$$

Parameters $\boldsymbol{a}$ and $\boldsymbol{m}$ are determined by

$$\frac{q}{G_p}=at^{-m} \tag{3-18}$$

Where $\boldsymbol{q}$ is the flow rate (volume/time), $\boldsymbol{a}$ is intercept (log-log plot of $\frac{\boldsymbol{q}}{\boldsymbol{G_p}}$ vs. $\boldsymbol{t}$), $\boldsymbol{G_p}$ is cumulative gas production. A plot of $\boldsymbol{q}$ against $\boldsymbol{t\ (a,\ m)}$ should also show a straight line with a slope of $\boldsymbol{q_1}$ and intercept of $\boldsymbol{q}_{\infty}$:

$$t(a,m)=t^{-m}\exp\left[\frac{a}{1-m}(t^{1-m}-1)\right] \tag{3-19}$$

It should be noted that depending on the operating conditions $\boldsymbol{q}_{\infty}$ might be positive, zero, or negative. Cumulative gas production can be determined using $\boldsymbol{q}_{\infty}$ is equal to zero as:

$$G_p = \frac{q_1 t(a,m)}{a t^{-m}} \tag{3-20}$$

Duong evaluated this model's accuracy on several types of wells, including tight, dry, and wet gas. He also discovered that most shale models had ***a*** value between 0 and 3 and ***m*** values between 0.9 and 1.3. When compared to the power law and Arps model, this model produces a reasonable estimation of cumulative production.

3.2.6 Logistic Growth Model (LGM)

Growth function models have been successfully applied in several life, social, and economic sciences branches to explain subjects such as population, cells, tumors, diseases, and market growth. The theory has strong connections to diffusion phenomena, providing a unifying framework for generating further insights into the decline curve analysis area in reservoir engineering. Growth functions provide a compact representation to account for the evolution of cumulative resources given confronting expansion and restraint forces. This phenomenon can be mathematically described by the solution of the following equation [20]:

$$Q(t) = \frac{k t^n}{(a + t^n)^2} \tag{3-21}$$

where ***Q*** is the cumulative production, ***t*** is the time, ***a*** is constant, ***n*** is the hyperbolic exponent, and ***k*** is the carrying capacity. Besides, the production rate can be described by the following formula:

$$q(t) = \frac{dQ}{dt} = \frac{knbt^{n-1}}{(a + t^n)^2} \tag{3-22}$$

The Advantages of the LGM can be given as [20]:

1- Easy to use, and very capable of being a positive data trend of existing products and providing a reasonable estimate of future production
2- Logistics models can very accurately assess the downward trend in behavior in a well
3- The ultimate recovery value that is expected can be predicted with this model and is also slightly simpler than the Arps model
4- The logistical growth model provides a new alternative method to forecast production in unconventional reservoirs empirically

The reported disadvantage of the LGM [20] is that it cannot predict non-physical values.

3.2.7 Extended Exponential DCA Model

Zhang et al. [21] combined the exponential form of the decline equation with a newly proposed empirical equation to calculate $\boldsymbol{D_i}$:

$$D_i = \beta_l + \beta_e e^{-t^n} \tag{3-23}$$

where $\boldsymbol{\beta_l}$, $\boldsymbol{\beta_e}$ are constants and, once calibrated using available production data, can capture both the early and late production profiles.

3.3 Limitations of Decline Curves

Although DCA is a popular technology in the industry, it has many shortcomings in shale wells [5]. As discussed, researchers have come up with interesting techniques to overcome some of the well-known shortcomings of DCA; however, many facts remain that make the use of DCA suboptimal.

One of the major criticisms of DCA is its lack of sensitivity to physical phenomena in shale wells that have to do with the fluid flow, the hydraulic fracture, and the reservoir characteristics. In cases like Marcellus and Utica shale reservoirs where short production periods are available, the use of DCA becomes increasingly problematic [5].

Shale wells are completed with different practices, a combination of long laterals with massive hydraulic fractures, which are the primary production driver. Completion design parameters introduce a new set of complexity to production behavior that cannot be readily dismissed, and they are entirely ignored and overlooked in DCA [5]. Recent production and completion data from multiple shale plays shows that such assumptions are invalid and quite costly for the operators.

3.4 Rate Transient Analysis (RTA)

Rate transient analysis (RTA) is the science of analyzing production data (flow rates and pressures). This method is an essential tool to estimate the reserve of oil and gas. It can also be defined as a modern DCA method [22]. Traditional DCA deals with a pseudo-steady-state flow regime and is an empirical / semi-empirical method. However, RTA deals with transient state flow, and the RTA flow equations have an analytical origin. RTA is known to give convincing estimates of reservoir parameters when low frequency (weekly or monthly) production data is available.

RTA has the same problems as numerical reservoir simulation since almost all of its approaches, especially when it forecasts production, mimic numerical modeling [5].

4 DATA-DRIVEN BASED FORECASTING

4.1 Introduction

Various decline curve models have been proposed for unconventional wells. However, it is hard to determine which model is more appropriate. Sometimes multiple models may fit a dataset equally well but provide different forecasts, and hastily selecting a model for probabilistic DCA can underestimate the uncertainty in a production forecast. Machine learning is changing the oil and gas industry by utilizing computing power more effectively and efficiently. Compared with the traditional DCA, machine learning is model-free (nonparametric) as the pre-determination of decline curve models is not required [23]. Using conventional techniques, human cognition can no longer discern important information when the amount of data increases. Data mining and machine learning techniques must be used to deduce information and knowledge from the raw data that resides in the databases [5].

Since our understanding of the physics and mechanics of shale storage and transport phenomena is quite limited, a data-driven model can play a vital role in extracting much-needed value from the data collected during the development of shale assets. This method can encompass any data-driven techniques, workflow, and solutions that attempt to increase recovery and production efficiency from shale plays [5]. Unlike conventional techniques such as Rate Transient Analysis (RTA) and numerical simulation that are heavily dependent on soft data such as fracture half-length, fracture height, fracture width, and fracture conductivity, this technique concentrates on using hard data (field measurements) in order to accomplish all its tasks [5].

An accurate predictive model will help us answer what will happen in a producing well or field and what we should do about it. This depends on the type of the algorithm and how the models are

fitted to the data, and how uncertainty is quantified. In this part, different machine learning algorithms will be reviewed.

4.2 Machine Learning

Machine learning (ML) involves algorithms that can learn from examples and experiences. It is based on the idea that some patterns exist in the data that were identified and used for future predictions. The difference from hardcoding rules is that the machine learns to find such rules without human intervention [24].

As shown in Figure 4-1, traditional programming differs significantly from ML. In traditional programming, a programmer create all of the rules based on a logical basis, and then the machine then executes the output based on the logical statement. More rules must be created as the system becomes more complicated. It can quickly become unsustainable to maintain. In ML, the machine learns how the input and output data correlate and then generates a a rule. The programmers do not need to develop new rules every time new new data is added. To improve efficacy over time, the algorithms adjust in response to new data and experiences [24].

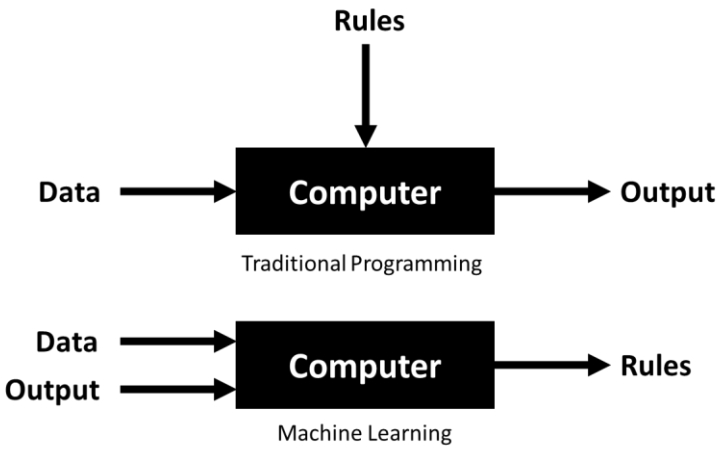


Figure 4-1: Traditional programming vs. machine learning (adapted from [24])

The core objectives of ML are learning and inference. First, the machine learns through the data to discover the patterns. The list of attributes that used to solve a problem is called a feature vector. The machine uses some statistical algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage describes the data and summarizes it into a model (Figure 4-2).

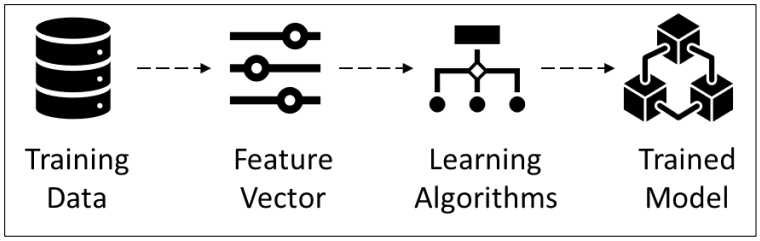


Figure 4-2: Learning phase in machine learning (adapted from [24])

When the model has been trained, we must put it in to the test using unseen data samples. The new data are transformed into a features vector, fed into the model, and provided with the prediction (Figure 4-3). The beauty of machine learning is that there is no need to retrai the model or update the rules. You can use the previously trained model to make inferences on new data.

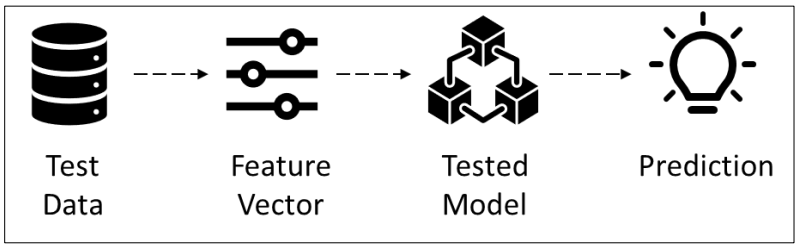


Figure 4-3: Inference from a model (adapted from [24])

ML can be categorized into two broad learning tasks: supervised and unsupervised. There are many other algorithms that reinforcement is the most important one (Figure 4-4). In the last one machine learns how to exposure to the environment.

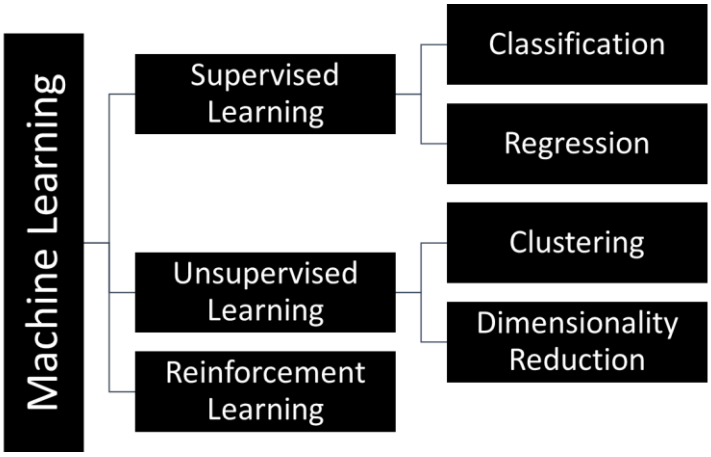


Figure 4-4: Machine learning classification (adapted from [24])

Supervised learning can be used when the output data is known. The algorithm will predict new data. There are two categories of supervised learning:

1- **Classification task**: probably is the most used supervised learning technique to predict categorical variables.
2- **Regression task**: in the machine learning field, it is often used to predict a continuous value and may predict the value of a dependent variable based on a collection of independent variables (also called predictors or regressors).

Table 3-1 is a list of some fundamental supervised learning algorithms. The training data in unsupervised learning is unlabeled, and the system tries to learn without a reference. It is useful when you do not know how to categorize the data, and you want the algorithm to detect patterns and classify the data. Reinforcement learning is a branch of ML in which systems are trained by receiving virtual rewards or penalties essentially in trial and error learning.

4.2.1 Key Machine Learning Definitions

Before getting started to discuss different ML algorithms, it is essential to have a clear understanding of some key terms and their definitions:

- **Model**: A ML model refers to a mathematical representation of a real-world problem. A model is the output of an ML algorithm run on data and represents what was learned by an

ML algorithm. To create a model, some training data are required to feed the ML algorithm to learn from [25].

- **Algorithm**: A ML algorithm refers to a procedure run on data to create an ML model. The algorithms perform pattern recognition and learn from data or are fit on a dataset [25]. Which kind of algorithm works best (supervised, unsupervised, classification, regression, etc.) depends on the kind of problem you are solving, the available computing resources, and the data's nature [26].
- **Training**: During training for machine learning, you feed an algorithm with training data. The learning algorithm looks for patterns in the training data that correlates the input parameters with corresponding targets. The output of the training process is a machine learning model, which you can then use to make predictions. This process also referred to as learning [27].
- **Parameter vs. Hyper-parameter**: A model parameter is a configuration variable that is internal to the model and may be estimated from data. In traditional machine learning literature, the model is a hypothesis and the parameters are how the hypothesis is fitted to a given dataset. A model hyperparameter, on the other hand, is a configuration that is external to the model and whose value cannot be derived from data. On a given problem, We cannot know what the optimum value for a model hyperparameter is. We can apply rules of thumb, copy values from other problems, or search for the optimum value by trial and error [28].
- **Regression**: The regression technique approximates a mapping function from input variables to a continuous output variable. A continuous output variable is a real value, such as an integer or floating-point value. These are often quantities, such as amounts and sizes [29].
- **Classification**: Classification is the task of approximating a mapping function from input variables to discrete output variables. Labels or categories are terms used to describe the output variables. The mapping function determines which class or category a given observation belongs to [29].

- **Clustering**: Clustering is a task which requires unsupervised machine learning. It involves automatically discovering natural grouping in data. Unlike supervised learning, clustering algorithms only interpret the input data and find natural groups or clusters in feature space [30].
- **Target**: The target is whatever the output of the input variables is. In case of a classification problem, it might be the individual classes that the input variables may be mapped to or the output value range in the case of a regression problem. If the training set is considered, then the target is the training output values that will be considered [27].
- **Feature**: Features are individual independent variables that act as the input in the system. Prediction models employ features to make predictions. Using a method called feature engineering, new features can also be obtained from existing ones. More simply, one column of a dataset is one feature. Sometimes these are also called attributes. Moreover, the number of features is called dimensions [27].
- **Label**: The final output is label. The output classes can alternatively be labels. Labeled data refers to groups of samples that have been tagged to one or more labels [27].
- **Overfitting**: An essential consideration in machine learning is how well the target function trained using training data generalizes to new data. Generalization works best if the signal or the sample that is used as the training data has a high signal-to-noise ratio. If that is not the case, the generalization would be poor, and we will not get good predictions. A model is overfitting if it fits the training data too well and there is a poor generalization of new data [27].
- **Regularization**: Regularization is the method to estimate a preferred complexity of the machine learning model so that the model generalizes and the over-fit/under-fit problem is avoided. This is done by adding a penalty on the different model parameters, thereby reducing the freedom of the model [27].

Table 4-1: Fundamental supervised learning algorithms [24]

Algorithm Name	Description	Type
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that is used for classification tasks. The output variable is binary rather than continuous	Classification
Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes until a final decision output is made	Regression Classification
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine (SVM) is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divides the classes. It is best used with a non-linear solver.	Regression Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest often generates simple decision trees and uses the 'majority vote' method to decide which label to return. For the classification task, the final prediction will be the one with the most vote, while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification
AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	The gradient-boosting tree is a state-of-the-art classification or regression technique. It focuses on the error committed by the previous trees and tries to correct it.	Regression Classification

Table 4-2: Unsupervised learning algorithms [24]

Algorithm	Description	Type
K-means clustering	Puts data into some groups (k) that each contains data with similar characteristics	Clustering
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters)	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system.	Clustering
Recommender system	Help to define the relevant data for making a recommendation	Clustering
PCA	They are mainly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

4.2.2 Linear Regression

Linear regression is a statistical technique for predicting future values based on past values. It can help in the detection of underlying trends and deal with cases involving overstated [31].

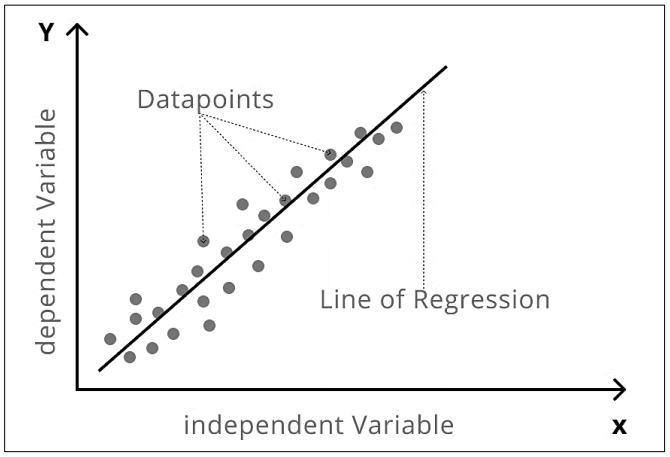


Figure 4-5: Linear Regression [31]

This regression type allows you to:

- Predict trends and future values through data point estimates.
- Forecast impacts of changes and identify the strength of the effects by analyzing dependent and independent variables.

We can represent functions/hypotheses $\boldsymbol{h}$ in a computer in different ways. The first intention would be to examine the linear regression model:

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x \qquad (4\text{-}1)$$

Here, the $\boldsymbol{\theta}i$'s are the parameters (also called weights) parameterizing the space of linear functions mapping from **X** to **Y** [32].

One reasonable method to obtain the parameter $\boldsymbol{\theta}$ is to make $\boldsymbol{h}(\boldsymbol{x})$ close to $\boldsymbol{y}$. To formalize this, we will define a function that measures, for each value of the $\boldsymbol{\theta}'s$, how close the $\boldsymbol{h}(\boldsymbol{x}(\boldsymbol{i}))'\, s$ are to the corresponding $\boldsymbol{y}(\boldsymbol{i})'\, s$. We define the cost function:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \tag{4-2}$$

The intuition would be to choose $\boldsymbol{\theta}$ to minimize the cost function. In (4-2), 2 in denominator just makes the math a bit easier and does not change the result [32].

The gradient descent algorithm is a search algorithm that sually used to minimize the cost function. It repeatedly changes $\boldsymbol{\theta}$ to make $\boldsymbol{J(\theta)}$ smaller [32]:

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta) \tag{4-3}$$

To implement this algorithm, we have to work out what is the partial derivative term on the right-hand side of the above equation [32]:

$$\theta_j := \theta_j + \alpha\sum_{i=1}^{m}\left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right)x_j^{(i)} \;\; (for\ every\ j) \tag{4-4}$$

4.2.3 Artificial Neural Network

Due to the complexity of governing conditions in subsurface shale reservoirs, simply a linear model can not satisfy this complexity, and we need more advanced algorithms. Neural Network (NN) can mimic the human brain to build these complex structures and determine the hidden relation between input parameters affecting the production. A set of different features acts as neurons in the network under the function of $\boldsymbol{f(x)}$ to output the production rate. NN stack all the neurons together such that one neuron passes its output as input into the next neuron, resulting in a more complex function (Figure 4-6). Since the information propagates in the forward direction through the network, then we refer to such a network as a feedforward neural network.

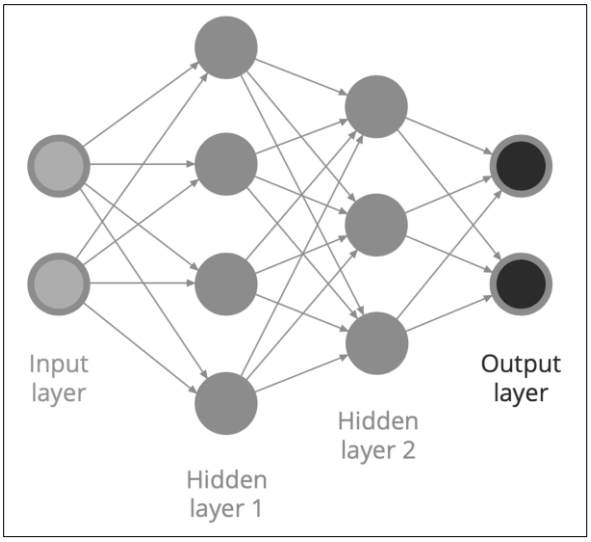


Figure 4-6: Sketch of a neural network design

Let us assume a particular artificial neuron (or perceptron) receive $\boldsymbol{n}$ inputs as $[\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}]$ and each connection carries a corresponding weight as $[\boldsymbol{w_1}, \boldsymbol{w_2}, \dots, \boldsymbol{w_n}]$. The first operation that is carried out multiplies the input values by their corresponding weight, and adds a bias term, $\boldsymbol{b}$, to their sum, producing an output, $\boldsymbol{z}$ [33]:

$$z = (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n) + b \tag{4-5}$$

We can alternatively represent as

$$z = \left(\sum_{i=1}^{n} x_i w_i\right) + b \tag{4-6}$$

Hence, a second operation is performed by each neuron that transforms the weighted sum by the application of a nonlinear activation function, $\boldsymbol{a(.)}$:

$$output = a(z) = a\left(\left(\sum_{i=1}^{n} x_i w_i\right) + b\right) \tag{4-7}$$

Here are examples of possible activation functions:

$$g(z) = \frac{1}{1 + e^{-z}} \ (sigmoid) \tag{4-8}$$

$$g(z) = \max(z, 0) \quad (ReLU) \tag{4-9}$$

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (tanh) \tag{4-10}$$

The sigmoid or logistic function (Figure 4-7) will mainly be used when we predict the probability as an output because it exists between 0 to 1 (S-shaped) [34].

Tanh or hyperbolic tangent function is like logistic sigmoid, but it ranges from -1 to 1 (s-shaped). The benefit is that on the tanh graph, the negative inputs will be mapped strongly negative, while zero inputs will be mapped near zero. The tanh function is mostly used to classify two classes [34].

ReLU or Rectified Linear Unit function is the most used activation function. It is used in almost all convolutional neural networks or deep learning and change from 0 to infinity. However, the issue is that all the negative values become zero immediately, which decreases the model's ability to fit or train from the data correctly [34].

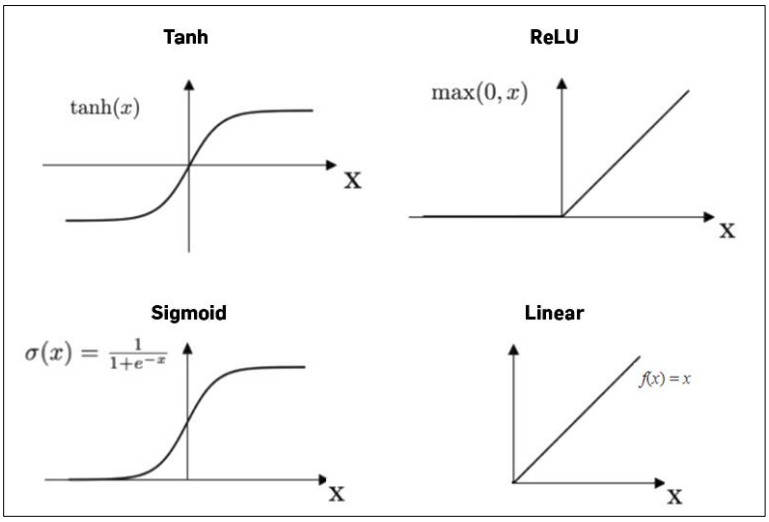


Figure 4-7: Common activation functions [35]

Figure 4-8 shows the operation performed by each neuron. If we had to integrate the bias term into the total as another weight, $\boldsymbol{w_0}$, We might describe the process done by each much more compactly:

$$y = a(z) = a\left(\sum_{i=0}^{n} x_i w_i\right) \tag{4-11}$$

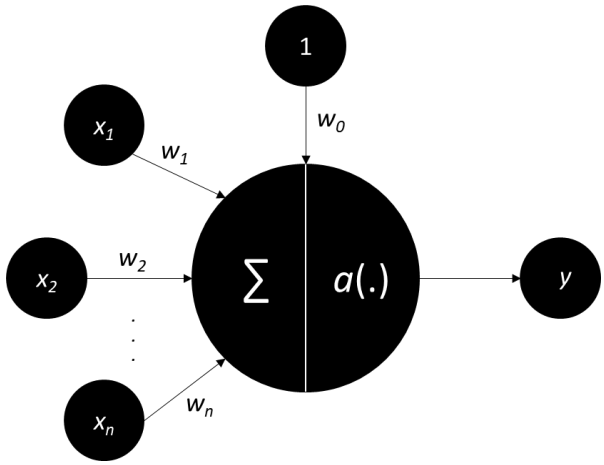


Figure 4-8: Nonlinear function implemented by a neuron

4.2.4 Support Vector Machine (SVM)

Support Vector Machine (SVM) [36] is a widely used supervised ML algorithm that finds its applications in problems pertaining to classification and regression. For a given set of training data, this algorithm determines a single or a group of hyperplanes in a space having infinite dimensions that can classify the data. Out of several hyperplanes, the best one is chosen in such a way that it denotes the farthest distance of separation from the nearest point in the training dataset of any class. This separation is termed as functional margin, and generally, if the margin becomes large, the generalization error (an assessment of the predictive accuracy of an algorithm when applied to formerly unseen data) of the classifier gets reduced. This technique is efficacious in high-dimensional spaces where the data is modeled based on several attributes. The decision function is composed of a subgroup of training data points, termed as support vectors, that stay nearer to the hyperplane and have a major impact on its position as well as orientation. With the help of these support vectors, the classifier's margin gets maximized. Apart from carrying out linear classification, SVM algorithms (Figure 4-9) can execute non-linear classification with the

help of the kernel method as the inputs are indirectly mapped into high-dimensional distinctive attribute spaces. When features outnumber samples to a great extent, kernel functions must be chosen appropriately to evade overfitting issues, and the regularization term becomes critical.

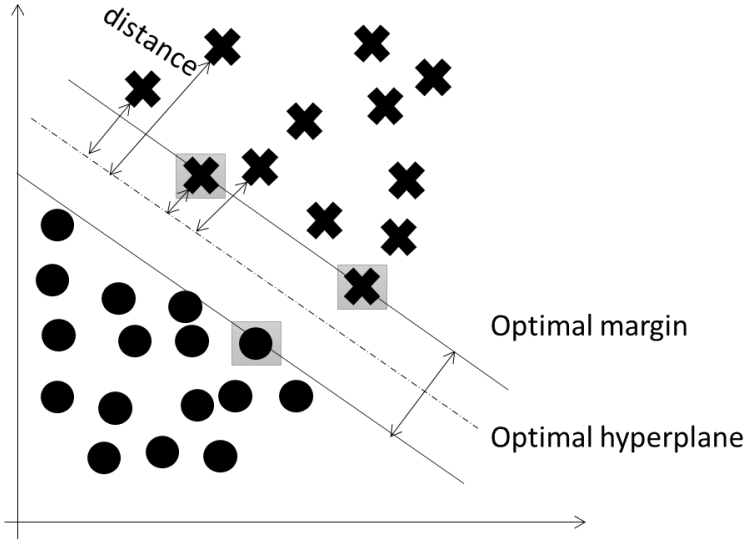


Figure 4-9: Support Vector Machine Algorithm. The support vectors, marked with grey squares, define the margin of largest separation between the two classes [36]

4.2.5 Boosting

The outcomes from different ML models for the same scenario and inputs may differ from each other. These individual models are weak learners. However, if we combine all the weak learners to work as one, the prediction would rely on input parameters. The accuracy of the model witll boost as the number of parameters increases. All boosting techniques, such as AdaBoost, Gradient Boosting, and XGBoost, are based on this logic.

Boosting is a technique that uses ML algorithms to combine weak learners and turn them into strong ones. Ensemble learning is used to improve the accuracy of a model [37]. There are two types of ensemble learning:

- **Sequential Ensemble Learning (Boosting)**

In the boosting technique, the weak learners' outputs associate sequentially during the training phase (Figure 4-10). The performance of the model is boosted by assigning higher weights to the samples that are incorrectly classified. An AdaBoost algorithm is an example of sequential learning.

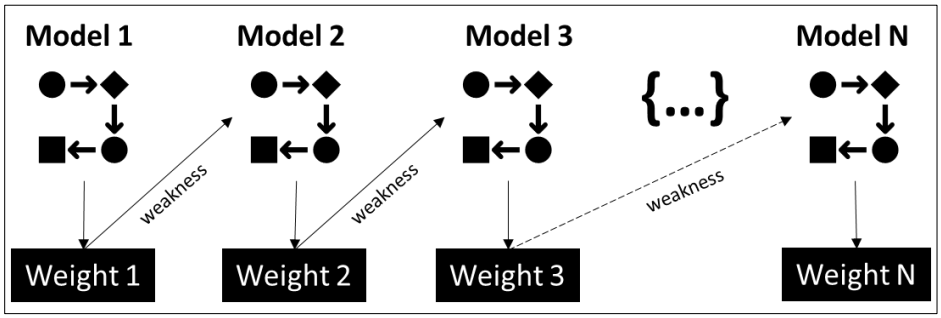


Figure 4-10: Sequential ensemble method (boosting) (adapted from [37])

- **Parallel Ensemble Learning (Bagging)**

In the bagging technique, the outputs from the weak learners are generated parallelly. It reduces errors by averaging the outputs from all weak learners (Figure 4-11). The random forest algorithm is an example of parallel ensemble learning [37].

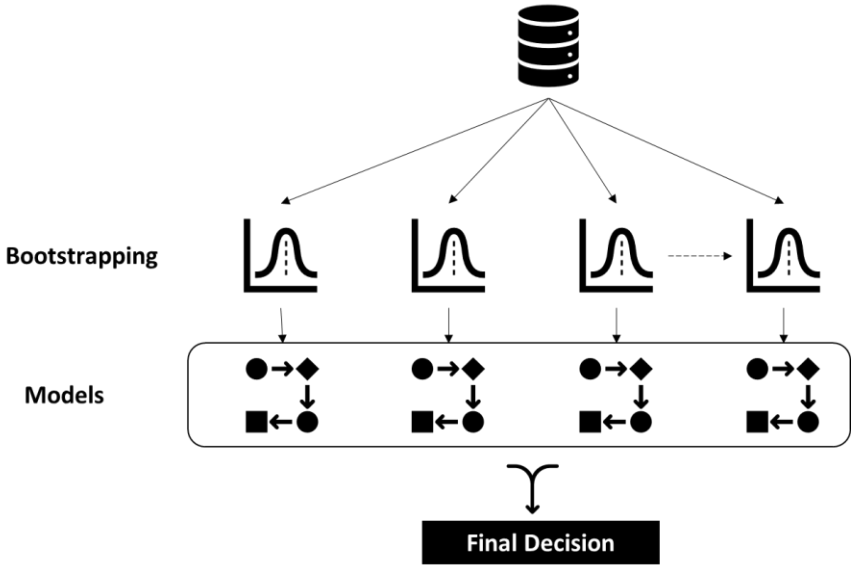


Figure 4-11: Parallel ensemble method (bagging) (adapted from [37])

4.2.5.1 Adaptive Boosting (AdaBoost)

Adaptive boosting is a technique used for binary classification. For implementing AdaBoost, we use short decision trees as weak learners. Here are steps for implementing AdaBoost [37]:

1. Train the base model using the weighted training data

2. Then, add weak learners sequentially to make it a strong learner

3. Each weak learner consists of a decision tree; analyze the output of each decision tree and assign higher weights to the misclassified results. This gives more significance to the prediction with higher weights.

4. Continue the process until the model becomes capable of predicting the accurate result

4.2.5.2 Gradient Boosting (GB)

Gradient boosting is a highly robust technique for developing predictive models. It applies to several risk functions and optimizes the accuracy of the model's prediction. It also resolves multicollinearity problems where the correlations among the predictor variables are high. Gradient boosting machines have been successful in various applications of machine learning.

We use gradient boosting to solve classification and regression problems. It is a sequential ensemble learning technique where the performance of the model improves over iterations. This method creates the model in a stage-wise fashion. It infers the model by enabling the optimization of an absolute differentiable loss function. As we add each weak learner, a new model is created that gives a more precise estimation of the response variable [37].

4.2.5.3 XGBoost

XGBoost algorithm is an extended version of the gradient boosting algorithm. It is basically designed to enhance the performance and speed of an ML model. In the gradient boosting algorithm, there is a sequential computation of data. Due to this, we get the output at a slower rate.

This is where we use the XGBoost algorithm. It increases the model's performance by performing parallel computations on decision trees. XGBoost is much faster than the gradient boosting algorithm. It improves and enhances the execution process of the gradient boosting algorithm [37].

4.2.5.4 Random Forest (RF)

Random Forest (RF) is an algorithm that comprises a large number of individual decision trees that function as an ensemble (collectively to achieve better performance in prediction) [38]. This ensemble learning algorithm (Figure 4-12) constructs multiple binary decision trees, also termed as Classification and Regression Trees or CART, from the training data and the descriptive variables pulled out randomly from the whole training dataset. The model of the constructed tree and the predicted values obtained from it are diverse because of the difference in the learned dataset possessed by each tree. The dataset that is not used to build the decision trees (out-of-bag selection) will be utilized to validate the model. Each unit has a different number of out-of-bag selections in the whole decision tree forming the random forest, and the classified values during selection are foretold differently for each tree. The probability of prediction is estimated for an individual unit. Random decision trees rectify the decision trees' tendency to overfit the training dataset, and the prediction accuracy is boosted. When the size of the forest becomes large with a large number of trees, the misclassification rate (generalized errors) reaches a limit. Randomly built data from the whole learning dataset is utilized while recovering individual decision trees, which remain unaffected by the occurrence of noise as well as outliers.

Random forest is a more advanced approach that makes multiple decision trees and merges them. The random forest model produces more reliable forecasts by taking an average of all individual decision tree estimates [31].

A random forest can be used for both classification and regression tasks, but it also has limitations. The model may be too slow for real-time predictions when analyzing a large number of trees. If you have no information other than the quantity data about a product, this method may not be as valuable. In such cases, the time series approach is superior.

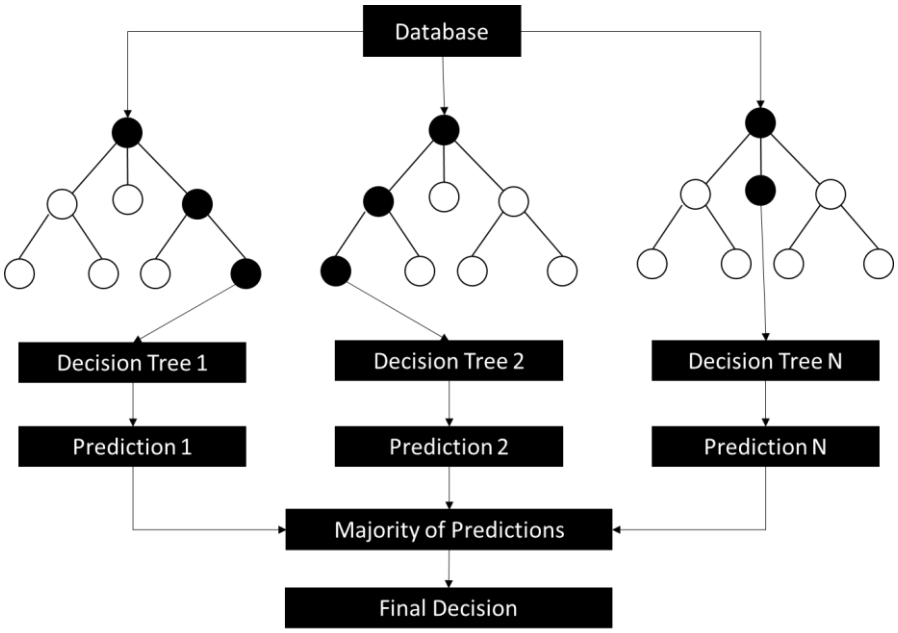


Figure 4-12: Random Forest algorithm

4.3 Machine Learning Models Development

Various steps are involved in developing a machine learning model. Following, we briefly explain these steps (Figure 4-13):

4.3.1 Data Collection and Preparation

In reality, oil and gas data are not in the form which can be directly fed to the ML model. Usually, they come from multiple sources and different formats. This part of the workflow is the most time-consuming and can lead to incorrect models with low prediction accuracy that cannot be fixed

easily. Therefore, great care must be taken in this initial part of the study to ensure that the data makes sense and that no bad data is fed into the model.

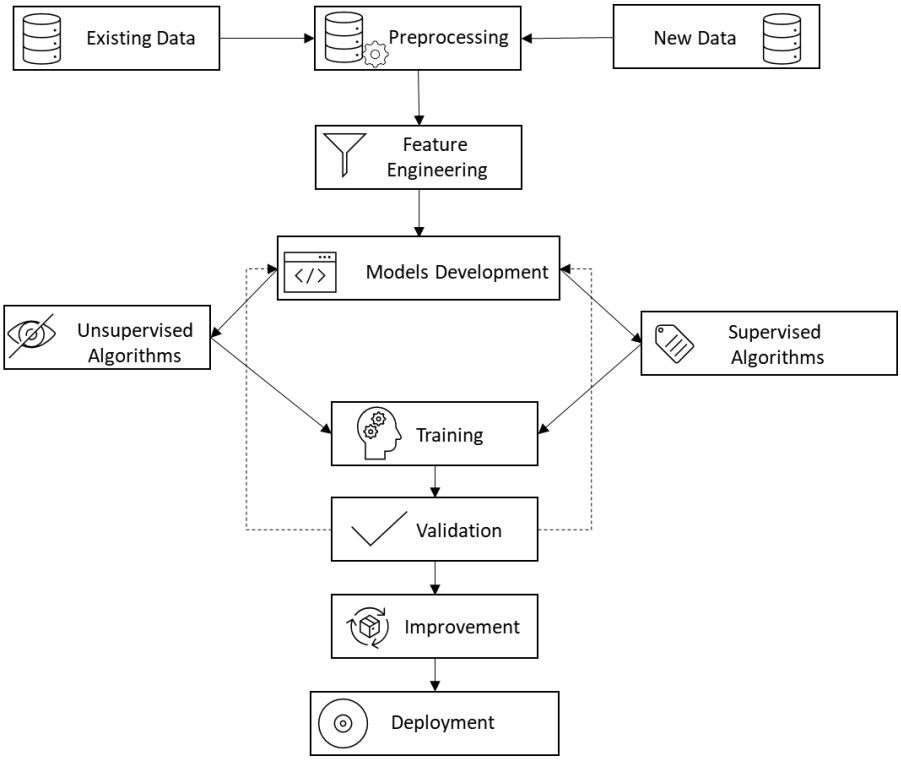


Figure 4-13: ML model development workflow

4.3.2 Exploratory Data Analysis

Exploratory data analysis is the first recommended step to understand the relationship between the predictors and the target. Exploratory data analysis, in general, can be a helpful tool when the importance of the predictors is not well understood at first glance. Besides the importance of the predictor, other interesting observations can be made by visualizing the data. It should be noted that there are various methods for exploratory data analysis ranging from univariate to multivariate for measuring the predictive power of input.

4.3.3 Feature Engineering

Feature engineering is the use of domain knowledge data and the creation of features that make machine learning models predict more accurately. It enables a deeper understanding of data and more valuable insights. Since feature engineering creates new features according to business goals, this approach is applicable in any situation where standard methods fail to add value. In custom ML modeling, a data scientist builds new features from existing ones to achieve higher forecast accuracy or to get new data.

Permutation feature importance is a model inspection technique that can be used for this purpose. It is defined to be the decrease in a model score when a single feature value is randomly shuffled [39]. It measures the increase in the model's prediction error after permuting the feature's values, which breaks the relationship between the feature and the outcome.

4.3.4 Training

Once the forecasting models are developed, it is time to start the training process with historical data to provide a ready-to-use trained model. In supervised learning, an ML algorithm builds a model by examining many examples and attempting to find a model that minimizes loss. Loss is the penalty for a bad prediction. The loss is a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. Training a model aims to find a set of weights and biases that have low loss, on average, across all examples. Figure 4-14 compare a high loss model with a low loss model.

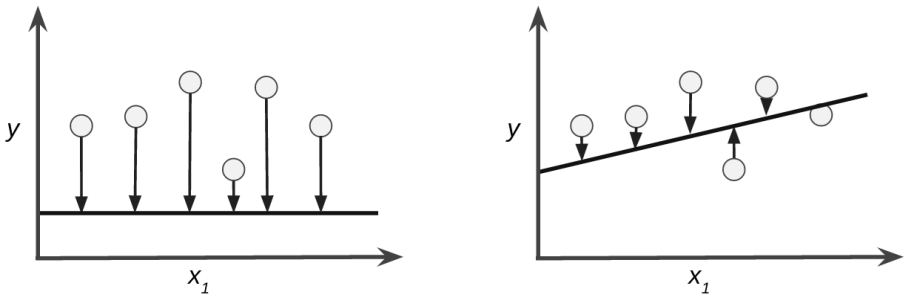


Figure 4-14: High (left) and low (right) loss - arrows represent loss, circles represent training data, and the solid lines represent predictions

To represent loss mathematically, we usually use a loss function that aggregates the individual losses. Squared loss (also known as L_2 loss) is a widespread loss function that examines a linear regression model as follow:

$$loss = \sum \big(observations - prediction(x)\big)^2 \tag{4-12}$$

Mean square error (MSE) is a commonly used criterion to evaluate the performance of the ML model. MSE is an averaged squared loss that calculates per individual example over the whole dataset and then divides by the number of examples:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y - \hat{y})^2 \tag{4-13}$$

For a large dataset with many features, we may need an iterative approach to reduce the loss. To simplify, consider a model with one input feature that returns one prediction as follow:

$$\hat{y} = b + w_1 x_1 \tag{4-14}$$

Here we need to find $\boldsymbol{b}$ and $\boldsymbol{w_1}$ such that we have the minimum loss. We start with an initial value and reevaluate iteratively to the point that overall loss stops changing or at least changes slowly. When this happens, we say that the model has converged.

Calculating the loss function for every conceivable value of $\boldsymbol{w_1}$ over the entire data set would be an inefficient way of finding the convergence point. Gradient descent presents an efficient

mechanism for this problem. The gradient descent algorithm then calculates the gradient of the loss curve at the starting point, which is equal to the derivative (slope) of the curve and has two characteristics: direction and magnitude. The gradient descent algorithm takes a step in the direction of the negative gradient in order to reduce loss as quickly as possible. The gradient descent algorithm adds some fraction of the gradient's magnitude to the starting point to determine the next point along the loss function curve. The gradient descent then repeats this process, edging ever closer to the minimum. Figure 4-15 shows the gradient descent process to reach the minimum loss.

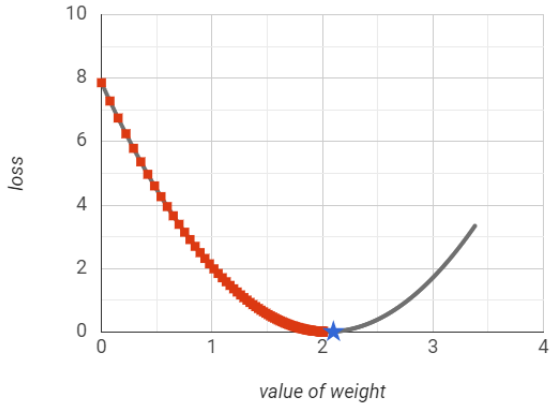


Figure 4-15: Gradient descent process

4.3.5 Validation

Validation is the process of making sure that the model generalizes well. Generalization is when the model is built using one set of data and performs well on a completely different set of data. The validation set is that other bucket of data on which we improve the generalization error. This step requires the optimization of the ML model parameters to achieve high performance. Every machine learning model has parameters and can additionally have hyper-parameters. Hyper-parameters are those parameters that cannot be directly learned from the regular training process. These parameters express higher-level properties of the model, such as its complexity or how fast

it should learn. While the objective of the training set is to learn parameters - such as the slope and intercept in the linear regression algorithm, the objective of the validation set is to tune the hyper-parameters [40].

4.3.5.1 Hold-Out Validation

In hold-out validation, the dataset splits randomly into training and validation sets (Figure 4-16). When using the hold-out method, as a rule of thumb, we can use 80% of data for training and the remaining 20% of the data for validation. This method is computationally cheap, and the training set is entirely independent; however, it does not generalize well [40].

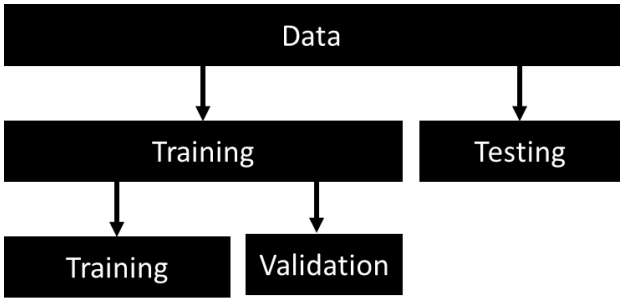


Figure 4-16: Hold-out validation (adapted from [40])

4.3.5.2 K-Fold Cross-Validation

In this method, data is split into ***K*** subsections called folds (Figure 4-17). *K*-fold validation evaluates the data across the entire training set by dividing it into ***K*** folds and then training the model ***K*** times, each time leave a different fold out of the training data and use it as a validation set. In the end, the performance metric (e.g., accuracy, ROC, etc.) is averaged across all ***K*** tests. Once the best parameter combination has been found, the model is retrained on the full data. This method usually generalizes well, but it is computationally expensive [40].

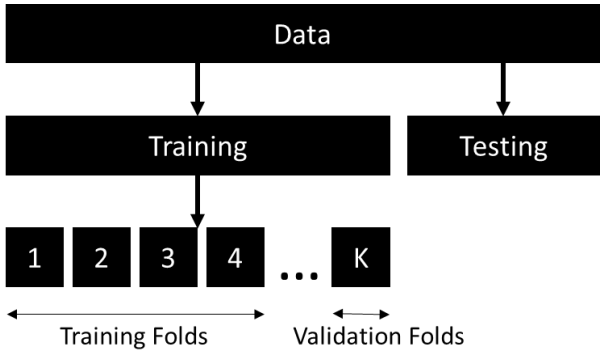


Figure 4-17: K-Fold cross-validation (adapted from [40])

4.3.6 Improvement

To find the best ML model, the usual practice is to develop several models with different levels of accuracy and then choose the ones that cover the business needs the best. The improvement step involves the optimization of analytic results. One way is to use model ensemble techniques and possibly reach a more accurate forecast. In this case, the accuracy is calculated by combining the results of multiple forecasting models.

Stacking or **Stacked Generalization** is an ensemble ML algorithm. This method uses a meta-learning algorithm to combine the predictions models of different ML algorithms effectively. Stacking has the ability to use a range of different well-performing models (either classification and regression) and create a perfect model with better performance.

The architecture of a stacking model involves two main parts [41]:

- **Level-0 models (base models)**: Models that fit on the training data and whose predictions are compiled
- **Level-1 model (metamodel):** Model that learns how to combine the predictions of the base models best.

As shown in Figure 4-18, the meta-model trains based on the predictions made by base models on out-of-sample data. Base models make predictions and these predictions, along with the expected outputs, provide the input and output pairs of the training dataset used to fit the meta-model [41].

The most common way to prepare the training dataset for the meta-model is via k-fold cross-validation of the base models, where the out-of-fold predictions are used as the basis for the training dataset for the meta-model [41].

Base models are often complex and diverse and can be a range of models that make very different assumptions about how to solve the predictive modeling task, such as linear models, decision trees, support vector machines, neural networks, random forests, and more. In contrast, the meta-model is often simple and provides a smooth interpretation of the predictions made by the base models. As such, linear models like linear regression are often used as the meta-model for regression tasks (predicting a numeric value) and logistic regression for classification tasks (predicting a class label) [41].

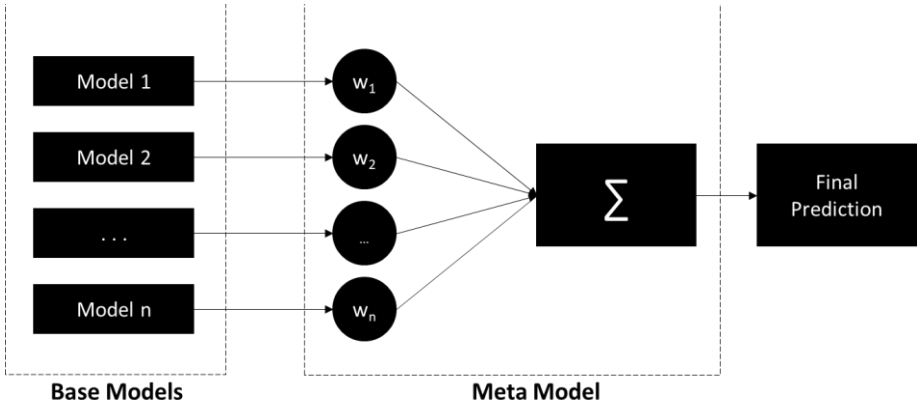


Figure 4-18: Stacked generalization (adopted from [42])

4.3.7 Deployment

After creating a reliable and accurate model through a systematic process, we can make forecasting for our problem. In this last stage of data-driven modeling, the ML model integrates and deploys into a production environment and can make practical decisions based on new data. For this case, we will set up a pipeline to aggregate new data to use in the ML model.

4.4 Challenges of Data-Driven Models

The primary challenge of machine learning is the lack of data or the diversity in the dataset. A machine cannot learn if there is no data available. Besides, a dataset with a lack of diversity gives the machine a hard time. A machine needs to have heterogeneity to learn meaningful insight. An algorithm can rarely extract information when there are no or few variations. It is recommended to have at least 20 observations per group to help the machine learn. This constraint leads to poor evaluation and prediction.

In non-physics-based problems like social networks and consumer relations, the relationship between correlation and causation cannot be resolved using physical experiments. On the other hand, in physics-based problems such as multi-phase flow in porous media (reservoir simulation and modeling), the interaction between parameters that are of interest have been understood and modeled by scientists. Therefore, treating the data as just numbers that need to be processed in order to learn their interactions is an over-simplification of the problem and hardly leads to valuable results. That is why many of such attempts have, at best, resulted in unattractive and mediocre outcomes [5].

5 TIME SERIES BASED FORECASTING

5.1 Introduction

Time series forecasting is a growing field of interest, playing an essential role in nearly all fields of science and engineering [43]. Unlike one-step ahead forecasting, multi-step ahead forecasting tasks are more complex since they have to deal with additional complications, like accumulation of errors, reduced accuracy, and increased uncertainty [44].

The forecasting domain has been influenced, for a long time, by linear statistical methods such as ARIMA models. However, it became increasingly clear that linear models are not adapted to many real applications. Several applicable nonlinear time series models were proposed in the same period, such as the bilinear, autoregressive threshold, and autoregressive conditional heteroscedastic (ARCH) models. Nowadays, Monte Carlo simulation or bootstrapping methods are preferred methods to compute nonlinear forecasts. However, the study of nonlinear time series analysis and forecasting is still in its infancy compared to the development of linear time series [45].

In the last two decades, machine learning models have drawn attention and have become serious contenders to classical statistical models in the forecasting community [43]. These models, also called black-box or data-driven models, are examples of nonparametric nonlinear models which use only historical data to learn the stochastic dependency between the past and the future. Artificial Neural Networks (ANNs), decision trees, support vector machines, and nearest neighbor

regression are used for modeling and forecasting nonlinear time series. Different strategies (or alternatives) have been proposed in the literature to tackle an H-step forecasting task. The *Recursive* strategy [46] iterates H times, a one-step-ahead forecasting model to obtain the $\boldsymbol{H}$ forecasts. After estimating the future series value, it is fed back as an input for the following forecast.

In contrast to the previous strategy, which uses a single model, the *Direct* strategy estimates a set of $\boldsymbol{H}$ forecasting models, each returning a forecast for the i-th value ($i \in \{1, \dots, H\}$).

A combination of the two previous strategies, called the *DirRec* strategy, has been proposed by Sorjamaa and Lendasse [47]. The idea behind this strategy is to combine aspects from both the *Direct* and the *Recursive* strategies. In other words, a different model is used at each step, but the approximations from previous steps are introduced into the input set.

In order to preserve, between the predicted values, the stochastic dependency characterizing the time series, the *Multi-Input Multi-Output* (*MIMO*) strategy has been introduced and analyzed [48]. Unlike the previous strategies where the models return a scalar value, the *MIMO* strategy returns a vector of future values in a single step.

The last strategy, called *DIRMO* [49], aims to preserve the most appealing aspects of *DirRec* and *MIMO* strategies. This strategy aims to find a trade-off between the property of preserving the stochastic dependency between the forecasted values and the flexibility of the modeling procedure.

5.2 Time Series

The time series forecasting is done using two strategies, namely the *recursive strategy* and the *direct strategy*. The recursive strategy includes Auto-Regressive Integrated Moving Average (ARIMA) and Neural Network Auto Regression (NNAR). The direct strategy is done with Extreme Gradient Boosting.

5.2.1 Recursive Strategy of Time Series Forecasting

The Recursive strategy [50] trains first a one-step model f

$$y_{t+1} = f(y_t, \dots, y_{t-n+1}) + w_{t+1} \quad (5\text{-}1)$$

With $\boldsymbol{t} \in \{\boldsymbol{n}, \dots, \boldsymbol{N-1}\}$ and then uses it recursively for returning a multi-step prediction. A well-known disadvantage of the recursive strategy is its sensitivity to the estimation error, since estimated values, instead of actual ones, are more and more used when we get further in the future, which is predicting the predictors. With the recursive strategy, the errors accumulated from the previous forecasts are part of the future forecasts because autoregression implies autocorrelation and dependency between consecutive forecasts [50].

5.2.1.1 ARIMA Model Baseline

In an autoregression model, we forecast the variable of interest using a linear combination of past values of the variable. The term autoregression indicates that it is a regression of the variable against itself [45]. Thus, an autoregressive model of order $\boldsymbol{p}$ can be written as

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \quad (5\text{-}2)$$

where $\boldsymbol{\varepsilon_t}$ is white noise, and $\boldsymbol{c}$ is the average of the changes between consecutive observations. This is like a multiple regression but with lagged values of $\boldsymbol{y_t}$ as predictors, we refer to this as an ***AR(p)*** model, an autoregressive model of order $\boldsymbol{p}$.

A moving average model uses past forecast errors in a regression-like model instead of using past values of the forecast variable in regression and can be written as

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \quad (5\text{-}3)$$

where $\boldsymbol{\varepsilon_t}$ is white noise. This is referred to as an ***MA(q)*** model, a moving average model of order $\boldsymbol{q}$. The values of $\boldsymbol{\varepsilon_t}$ are not observed, so it is not the usual practical regression found in normal applications.

The combination of an autoregression and a moving average model and the inclusion of reverse differencing term results in an ARIMA model. The ***I*** in ARIMA refers to integration which is the reverse of differencing. It can be written as

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \qquad (5\text{-}4)$$

where $\boldsymbol{y'_t}$ is the differenced series (it may have been differenced more than once). The predictors on the right-hand side include both lagged values of $\boldsymbol{y_t}$ and lagged errors [45]. This is called an **ARIMA(p,d,q)** model where p is the order of the autoregressive part, d is the degree of first differencing involved, and q is the order of the moving average part. ARIMA can be viewed as a linear regression where the predictors are a combination of the response variable's lagged variables, and the unobserved error's lagged variables.

5.2.1.2 Neural Network Autoregression

Artificial Neural Networks (ANN) are forecasting methods whose mathematical models are inspired by the mechanism of the brain. ANNs make it possible to model complex nonlinear relationships between the response variable and its predictors.

A non-linear combination of the inputs obtains the forecasts. The coefficients attached to these predictors are called weights. The weights are selected in the neural network framework using a learning algorithm that minimizes a loss function such as ***MEA*** (Mean Absolute Error), ***RMSE*** (Root Mean Squared Error), or others [45].

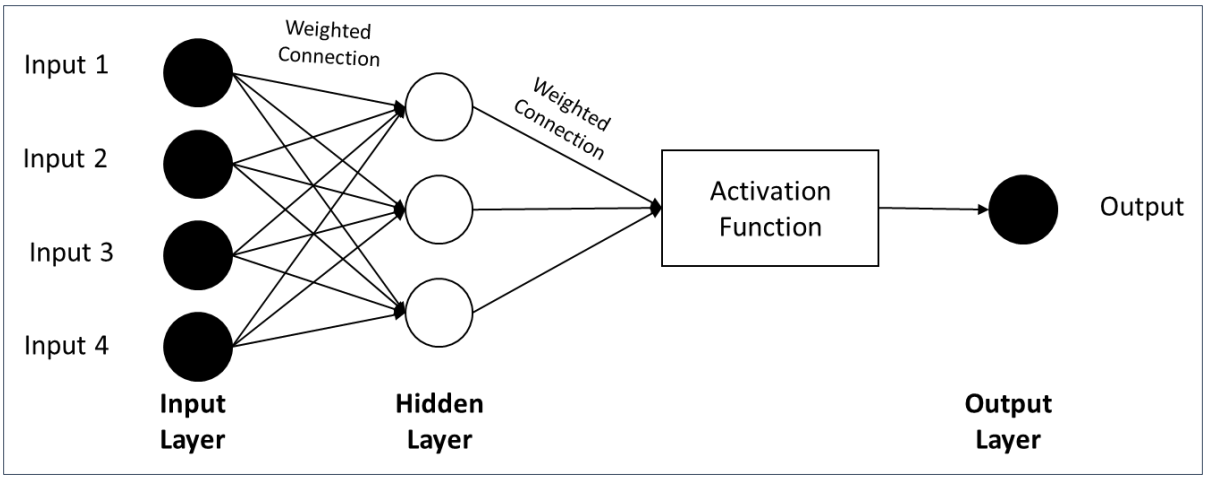


Figure 5-1: A neural network with four inputs and one hidden layer with three hidden neurons

the mathematical model for Figure 5-1 is

$$z_j = b_j + \sum_{i=1}^{4} w_{i,j} x_i \tag{5-5}$$

For the hidden layer, we can use a nonlinear function such as a sigmoid function to give the input for the next layer

$$s(z) = \frac{1}{1 + e^{-z}} \tag{5-6}$$

Suppose $\boldsymbol{p}$ denotes the number of lagged values of response variables that are used as inputs and $\boldsymbol{k}$ denotes the number of hidden nodes that are present, the output one-layer feed-forward neural network autoregression can be written as, if it non-seasonal

$$NNAR(p, k) \tag{5-7}$$

Or if it is seasonal,

$$NNAR(p, P, k) \tag{5-8}$$

where $\boldsymbol{P}$ denotes the number of seasonal lags, assume $\boldsymbol{y}$ is the response variable from the time series. Let $\boldsymbol{y_t}$ depend linearly and nonlinearly on the past values of $\boldsymbol{y}$. Example with $\boldsymbol{y}$ depends on the three previous values:

$$y_t = f(y_{t-1}, y_{t-2}, y_{t-3}) + \rho_1 y_{t-1} + \rho_2 y_{t-2} + \rho_3 y_{t-3} + \epsilon \quad (5\text{-}9)$$

where $\boldsymbol{f}()$ is a non-linear function given by a neural network (***MLP***); this is a combination of a classical ***AR*** model and neural network model. Let us denote all the lagged responses (features) as X.

$$y_t = f(X) + \rho_1 x_1 + \rho_2 x_2 + \rho_3 x_3 + \epsilon \quad (5\text{-}10)$$

If we assume that ***f(X)*** has a linear layer as the last layer, this is an example of a network with a skip-layer connection or residual network. Using more than one layer in ***f()***, then the model can be generalized as

$$f(x) = f_4\left(f_3\left(f_2\left(f_1(X)\right)\right)\right) \quad (5\text{-}11)$$

which is a network with four layers. If we add skip-layer connections:

$$f(x) = f_4(f_3(f_2(f_1(X), X), X), X) \quad (5\text{-}12)$$

All layers $\boldsymbol{f_i}$ $(\boldsymbol{f_{i-1}}, \boldsymbol{X})$ depend on the output from the previous layer and directly on the raw features X (which in this case is lagged responses).

5.2.2 Direct Strategy of Time Series Forecasting

The challenge is to avoid as much as possible predicting the predictors, in other words using the so-called recursive strategy of forecasting [51]. An alternative is the direct strategy of forecasting. The Direct strategy [51] learns $\boldsymbol{H}$ models independently $\boldsymbol{f_h}$

$$y_{t+h} = f_h(y_t, \dots, y_{t-n+1}) + w_{t+h} \quad (5\text{-}13)$$

with $\boldsymbol{t} \in (\boldsymbol{n}, \dots, \boldsymbol{N} - \boldsymbol{H})$ and $\boldsymbol{h} \in (\boldsymbol{1}, \dots, \boldsymbol{H})$ and returns a multi-step forecast by concatenating the $\boldsymbol{H}$ predictions. Since the direct strategy does not use any approximated values to compute the forecasts, it is not prone to any accumulation of errors. However, it has some disadvantages. Firstly, since the $\boldsymbol{H}$ models have learned independently, no statistical dependencies between the

predictions $\hat{y}_{N+h}$ is considered even if they truly exist. Secondly, direct strategy methods often require higher functional complexity than iterated ones in order to model the stochastic dependency between two series values at two distant instants. Thirdly, this strategy demands considerable computational time since the number of models to learn is equal to the size of the horizon.

Direct forecasting involves creating a series of distinct horizon-specific models to forecast each horizon of a time series [51]. The direct forecasting approached used in the algorithm includes the following steps (Figure 5-2) [52]:

1- Build a series of horizon-specific short-, medium-, and long-term forecast models
2- Assess model generalization performance across a variety of heldout datasets through time
3- Select those models that consistently performed the best at each forecast horizon and combine them to produce a single ensemble forecast

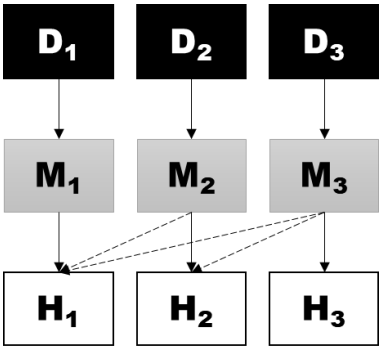


*Figure 5-2: Direct forecasting approach. **D** represents direct forecast; **M** represents model, and **H** represents the horizon [51]*

5.3 Cross-Validation

Cross-validation can be used to learn parameters or hyperparameters in order to avoid overfitting.

5.3.1 K-Fold Cross-Validation

K-fold cross-validation can be used to maximize the information obtained from one dataset. The procedure divides the dataset into K roughly equally sized pieces, where *K - 1* of the data folds are

used to train the model, and the last fold is used for validation of the obtained model [53]. The procedure is then repeated for all $\boldsymbol{K}$ folds, and the $\boldsymbol{K}$ performance measures are averaged to achieve a cross-validated model. For $\boldsymbol{k} \in (\mathbf{1}, \ldots, \boldsymbol{K})$, the cross-validated prediction error is

$$CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} M\left(y_i, \hat{f}^{-k(i)}(x_i)\right) \tag{5-14}$$

where $\hat{\boldsymbol{f}}^{-\boldsymbol{k}}(\boldsymbol{x})$ is the fitted model without the $\boldsymbol{k}^{\boldsymbol{th}}$ fold, and $\boldsymbol{M}(\boldsymbol{y}, \hat{\boldsymbol{f}}(\boldsymbol{x}))$ is the misclassification rate. When there is some tuning parameter/parameters to be fitted with the model, for instance, one or more penalty factors, the prediction error becomes

$$CV(\hat{f}, \lambda) = \frac{1}{N}\sum_{i=1}^{N} M\left(y_i, \hat{f}^{-k(i)}(x_i, \lambda)\right) \tag{5-15}$$

$\boldsymbol{CV}(\hat{\boldsymbol{f}}, \boldsymbol{\lambda})$ is then the estimate of the error curve, which is minimized $\hat{\boldsymbol{\lambda}}$, the optimal value of $\boldsymbol{\lambda}$.

5.3.2 Time Series Cross-Validation

Time-series cross-validation is done because there is a series of test sets, each consisting of a single observation. The corresponding training set consists only of observations that occurred prior to the observation that forms the test set. Thus, no future observations can be used in constructing the forecast [54]. Figure 5-3 illustrates the training and test sets series, where the blue observations form the training sets and the red observations form the test sets. As multi-step forecasts are preferred over one-step forecasts, we need to modify the cross-validation procedure to use multi-step errors. Suppose that we are interested in models that produce good 4-step-ahead forecasts. Then the corresponding diagram is shown in Figure 5-4. A good way to choose the best forecasting model is to find the model with the smallest ***RMSE*** computed using time series cross-validation [54].

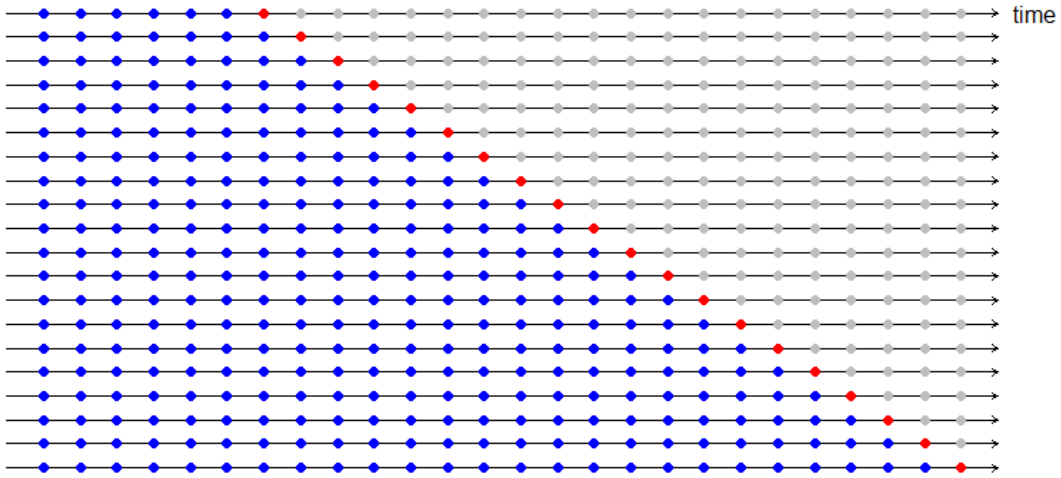


Figure 5-3: Time series cross-validation [54]

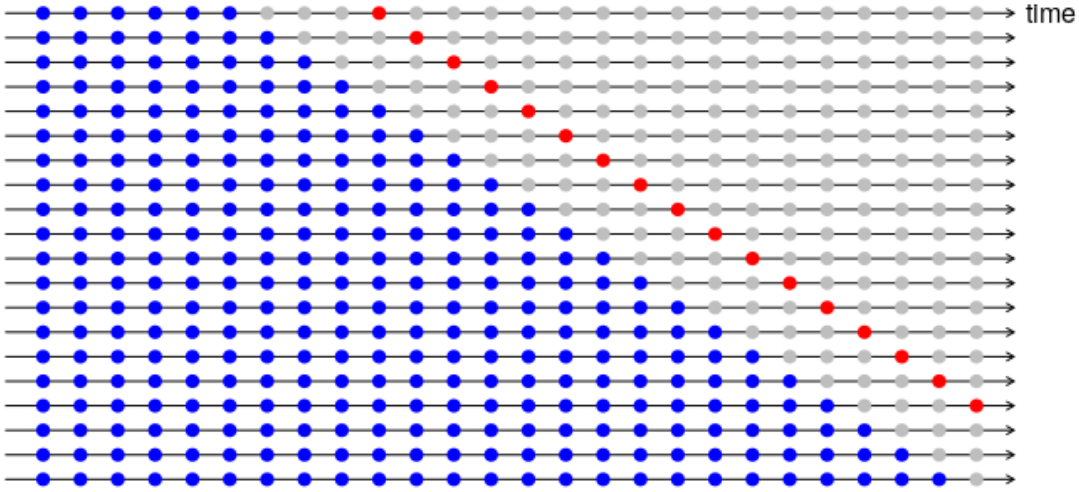


Figure 5-4: 4-step-ahead forecasts [54]

6 DATA ANALYSIS FOR DUVERNAY SHALE

6.1 Introduction

The Duvernay Shale is an emerging oil and liquids-rich gas formation in the Western Canada Sedimentary Basin (WCSB), which covers an area of approximately 130,000 square kilometers (km^2) of Alberta, or 20% of the province [55] (Figure 6-1). The formation was deposited during the Upper Devonian age (372 million years ago) and consisted of laminated bituminous shale and argillaceous limestone [56]. Tight gas and oil resources are trapped in the fine pores of the organic-rich mudstone rock and can be extracted through a combination of horizontal drilling and hydraulic fracturing. Duvernay resource is located near existing pipelines, including natural gas pipelines that connect natural gas markets within Canada and between Canada and the United States (US).

The productive areas of Duvernay have been subdivided into three assessment areas: Kaybob in the north, Edson-Willesden Green in the center, and Innisfail in the south. These assessment areas were divided based on similar geological characteristics and current development trends.

Industrially believe that the Duvernay is the closest analog to the Eagle Ford since they are both over-pressured reservoirs, and both formations contain volatile oil, condensate, wet gas, and dry gas windows that are all believed to be productive.

The Duvernay shale contains a massive initial resource in place; however, the amount that can be economically recovered depends on drilling and completions optimization, cost reductions, expected liquids yields, commodity pricing, and social, environmental, and regulatory constraints [55].

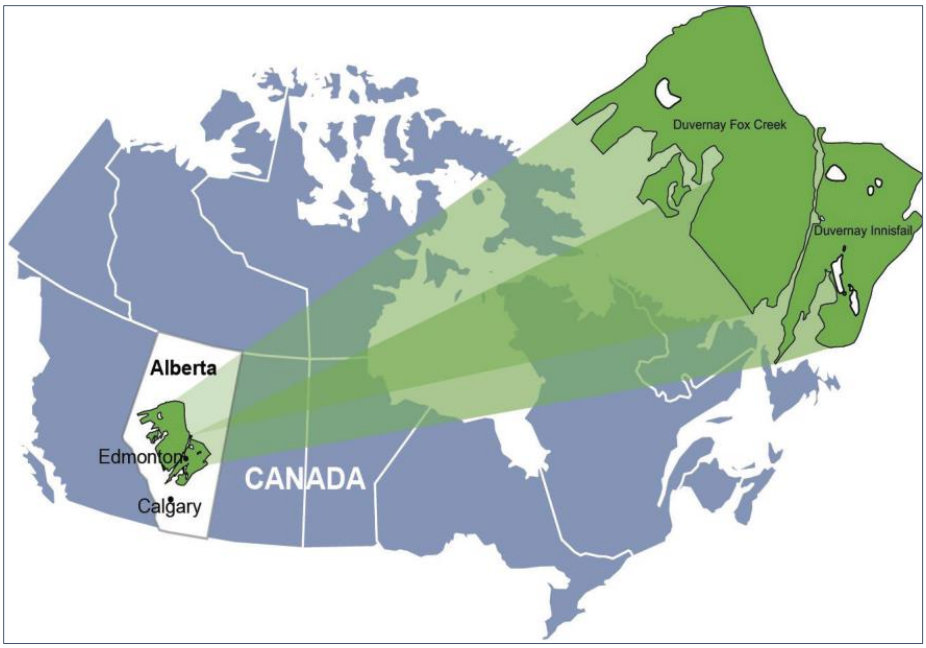


Figure 6-1: Duvernay depositional extent in central Alberta, Canada [55]

6.2 Petroleum Systems

According to Magoon (1988), a petroleum system includes all those geologic elements and processes that are essential for an oil and gas deposit to exist. These basic elements include source rock, maturation, migration path, reservoir rock, trap, and seal [57].

The Duvernay formation was deposited contemporaneously with several large Leduc reef complexes during the Frasnian age (382.7-372.2 Ma) and comprised a sequence of dark brown to black, bituminous, slightly argillaceous carbonates interbedded with gray-green, calcareous shales. These sediments are characteristically organic-rich, with total organic carbon (TOC) values of up to 20 percent. The Duvernay thickens northward and eastward up a depositional slope [58].

The Duvernay unit represents accumulations under marine, deep-water, low-energy, basinal conditions. Euxinic conditions are suggested by the absence of fauna, preservation of organic material (Type II oil-prone kerogen). Undoubtedly, anoxic conditions, combined with slow

sedimentation rates within this depositional basin, are the main reasons for preserving abundant organic material in this rich source rock [58].

The Duvernay occurs widely throughout Alberta Figure 6-2, although much of it is thermally immature. However, a broad arcuate band of the Duvernay is within the oil window, with the most deeply buried portion being overmature [58].

Duvernay-sourced oils are low in sulfur (less than 0.5 percent) and have pristane/phytane ratios of 1.5 to 2.4, recording source deposition in normal marine salinities in an oxygen-poor environment. Creaney and Allan (1990) outlined Duvernay-sourced oils have migrated extensively within the Western Canada Basin. Overmaturity in the western part of the WCSB has produced significant gas generation from Duvernay. In addition, high present-day temperatures have induced the reaction of interbedded anhydrite with hydrocarbon gases to form hydrogen sulfide, which, due to the lack of iron in these carbonate systems, has resulted in considerable reserves of sour gas [58].

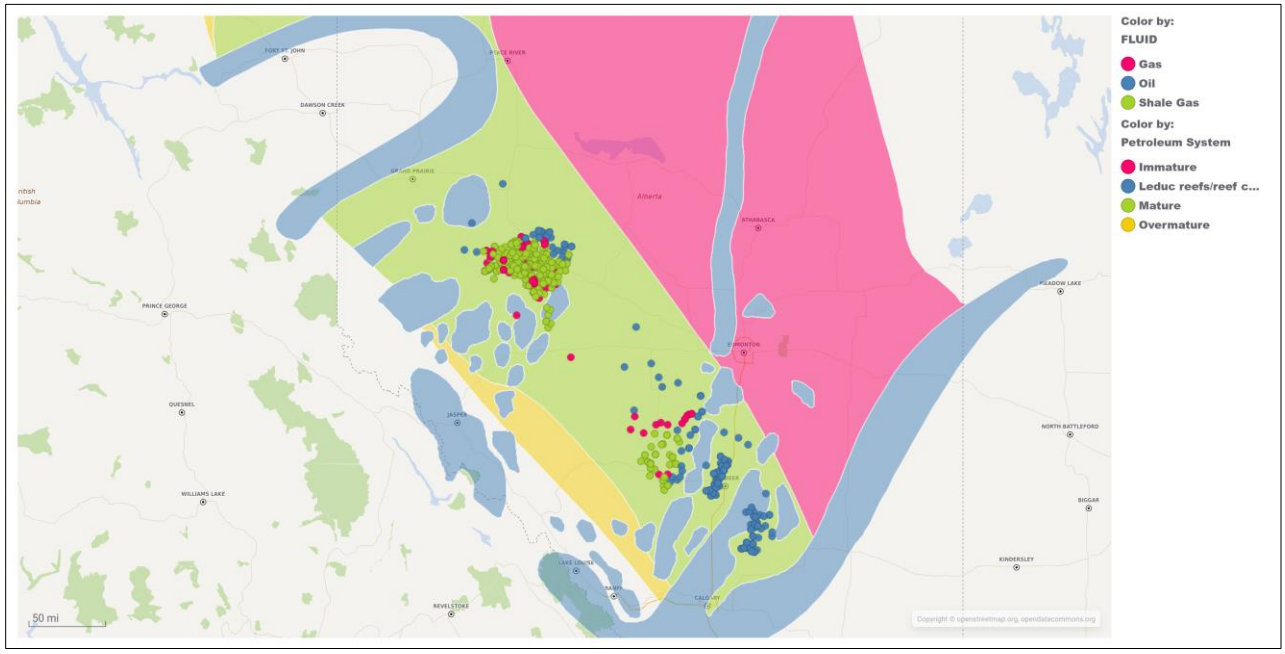


Figure 6-2: Petroleum Systems in Duvernay shale (data source [59])

It can be noticed from Figure 6-2 that the gas zone is under the oil zone, which is counter-intuitive. This upside-down fluid distribution can be explained by the formation of an overpressure regime in the deeper area of Duvernay. This area is in hydraulic isolation and is abnormally pressure with all the pore spaces filled with hydrocarbon. The thermal maturity and pressure increase from east to west of the basin that causes gas production to be dominant.

6.3 Development History

The Duvernay Formation was first defined by geological staff at the western division of Imperial Oil Limited (1950) for dark grey to brown, bituminous shales in wells drilled near the townsite of Duvernay east-central Alberta. The reference well in Alberta is the Anglo Canadian Beaverhill Lake No. 2 11-11-050-17W4M well [55].

Although the Duvernay Shale certainly has the potential to be a highly productive and economic play, its development has been prolonged up to 2011 (Figure 6-5). The majority of the production from the play as a whole is gas.

Figure 6-3 shows that most of the producer wells in Duvernay are shale gas type (63%). Figure 6-4 gives the gradual well development in the area. As it shows, major wells have been drilled from 2016 to 2019 due to increasing global oil and gas prices. Figure 6-5 also shows the active producer wells since 1980. 2019 had the highest producing wells in the area.

Figure 6-6 shows wells activity in the top 10 fields. Kaybob field is the most active region in the play. This region straddles the gassy and liquids-rich windows of the play. The development of Duvernay is mainly focused on the Kaybob region and the Willesden Green region in the southeast.

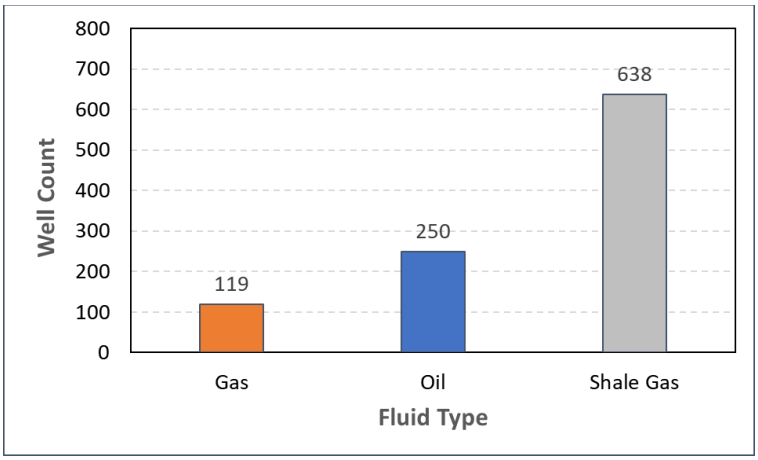
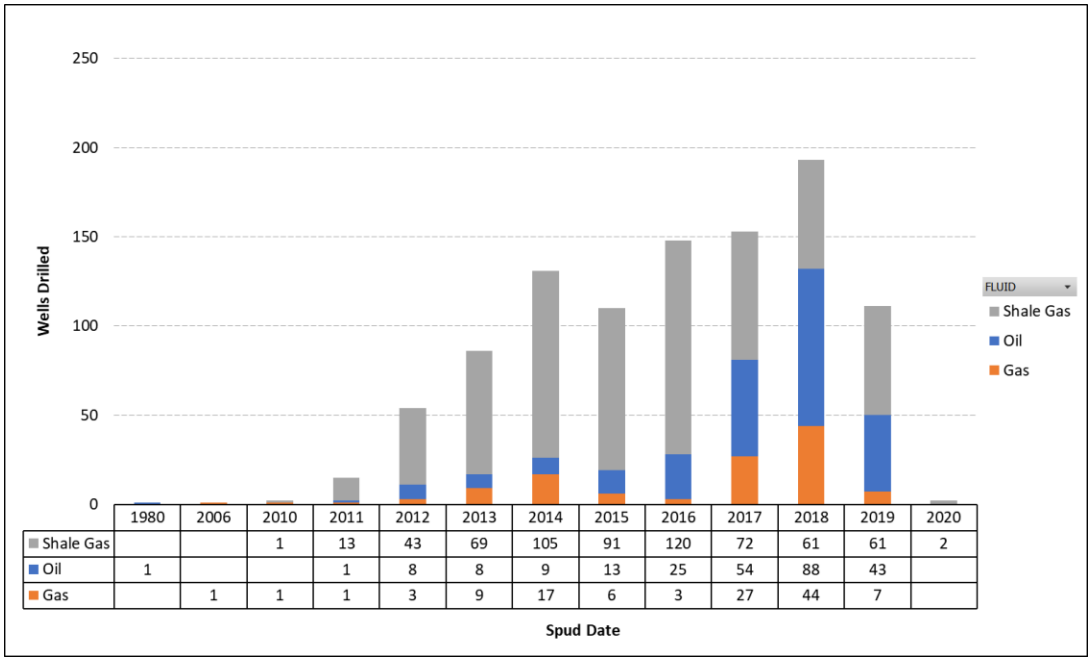
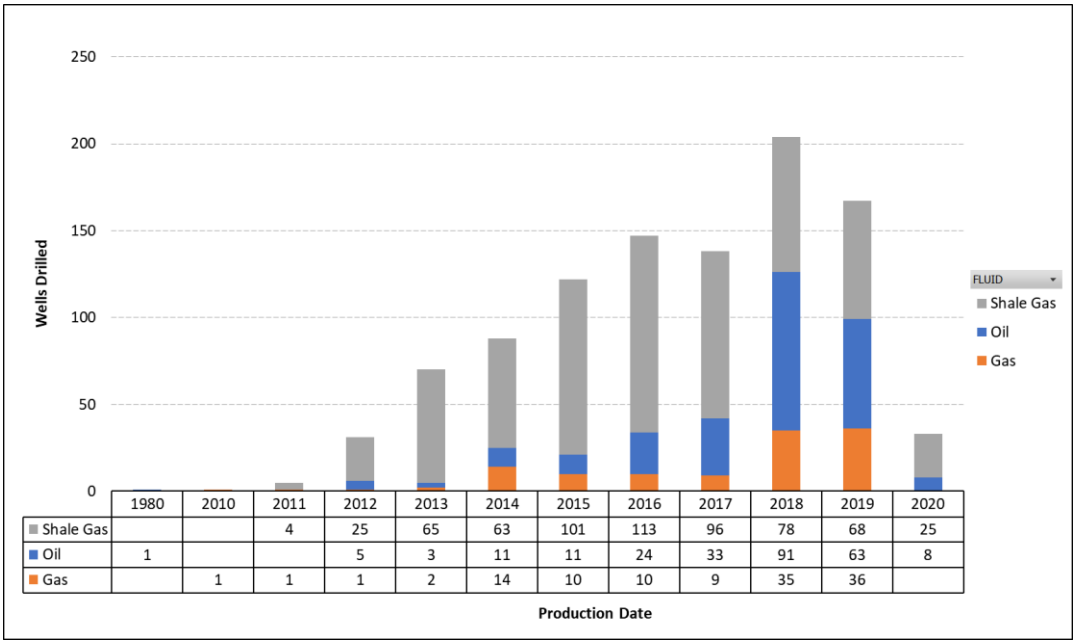


Figure 6-3: Fluid type of producer wells in Duvernay (data source [60])



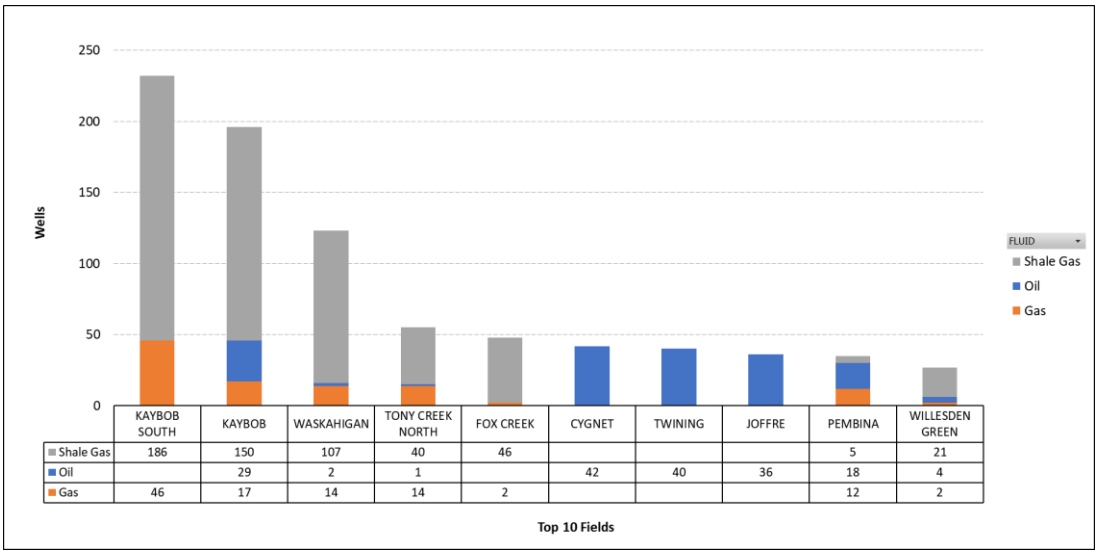
	1980	2006	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Shale Gas			1	13	43	69	105	91	120	72	61	61	2
Oil	1			1	8	8	9	13	25	54	88	43	
Gas		1	1	1	3	9	17	6	3	27	44	7	

Figure 6-4: Well development in Duvernay since 1980 (data source [60])



	1980	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Shale Gas			4	25	65	63	101	113	96	78	68	25
Oil	1			5	3	11	11	24	33	91	63	8
Gas		1	1	1	2	14	10	10	9	35	36	

Figure 6-5: Well producers count in Duvernay since 1980 (data source [60])



	KAYBOB SOUTH	KAYBOB	WASKAHIGAN	TONY CREEK NORTH	FOX CREEK	CYGNET	TWINING	JOFFRE	PEMBINA	WILLESDEN GREEN
Shale Gas	186	150	107	40	46				5	21
Oil		29	2	1		42	40	36	18	4
Gas	46	17	14	14	2				12	2

Figure 6-6: well activities in top 10 fields in Duvernay area (data source [60])

6.4 Completion

Well completion is an important process in shale plays due to the high CAPEX of drilling horizontal wells and multi-stage hydraulic fracturing. Figure 6-7 shows ***spm*** (perforation density and the number of perforation shots per meter) distribution in different wells. Most of the wells

have less than 4000 spm. Several deep wells have a huge number of perf and frac (more than 10,000 ***spm***). The production rate of these wells shows that the production will interestingly reduce after a few months of production. Figure 6-8 shows the frac stages in different wells across Duvernay shale. It seems most of the wells have 20-60 stages. Figure 6-9 and Figure 6-10 show the direct impact of these parameters on production rate. All of these figures show the dependency of production rate to the number of perforations and frac stages. This dependency for gas production is higher than oil production.

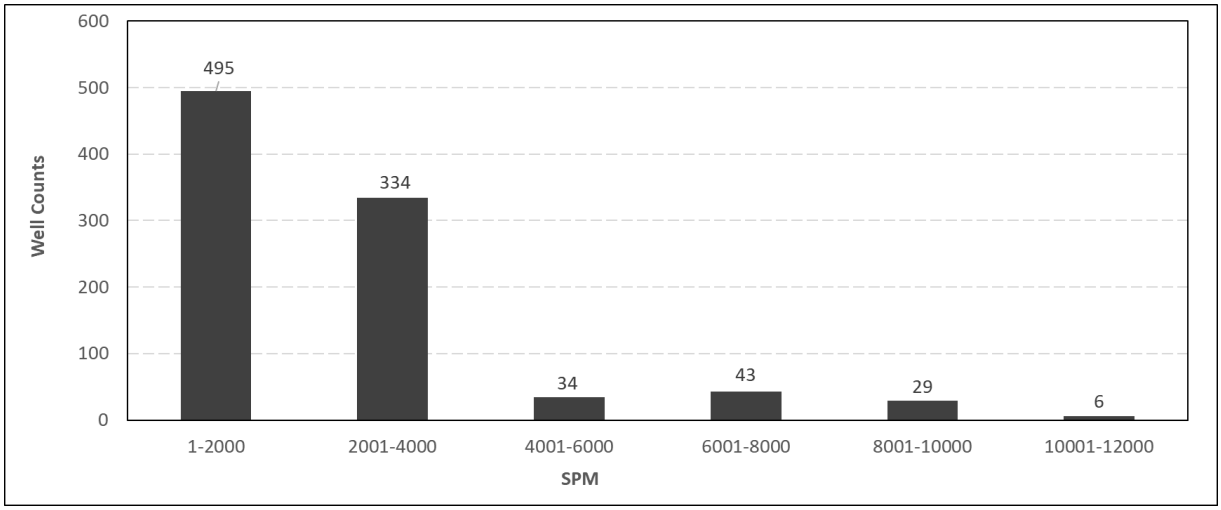


Figure 6-7: SPM distribution in Duvernay Wells (data source [60])

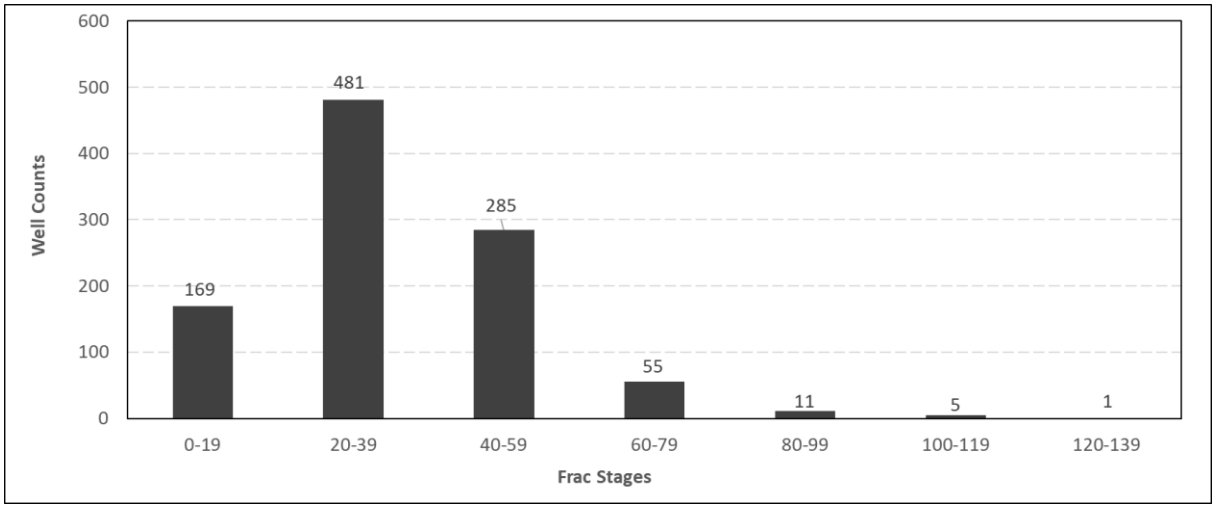


Figure 6-8: Frac stage distribution in Duvernay wells (data source [60])

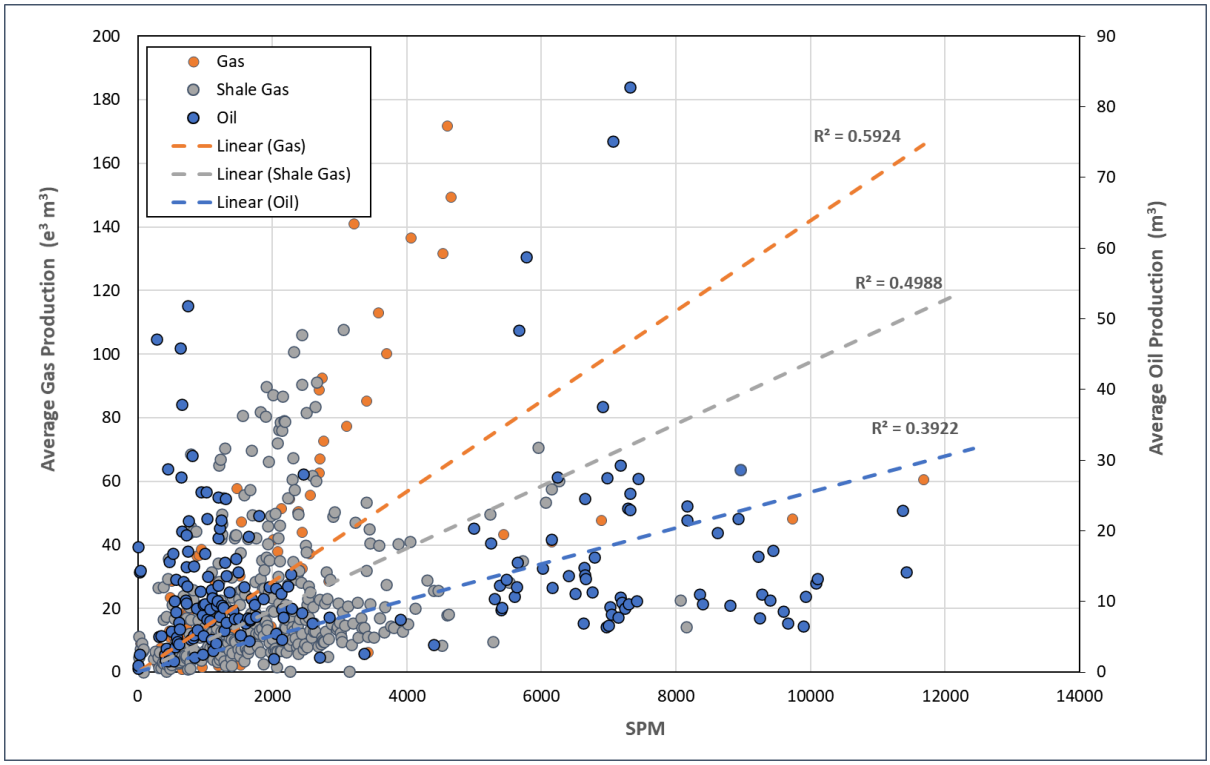


Figure 6-9: SPM vs. production rate (data source [60])

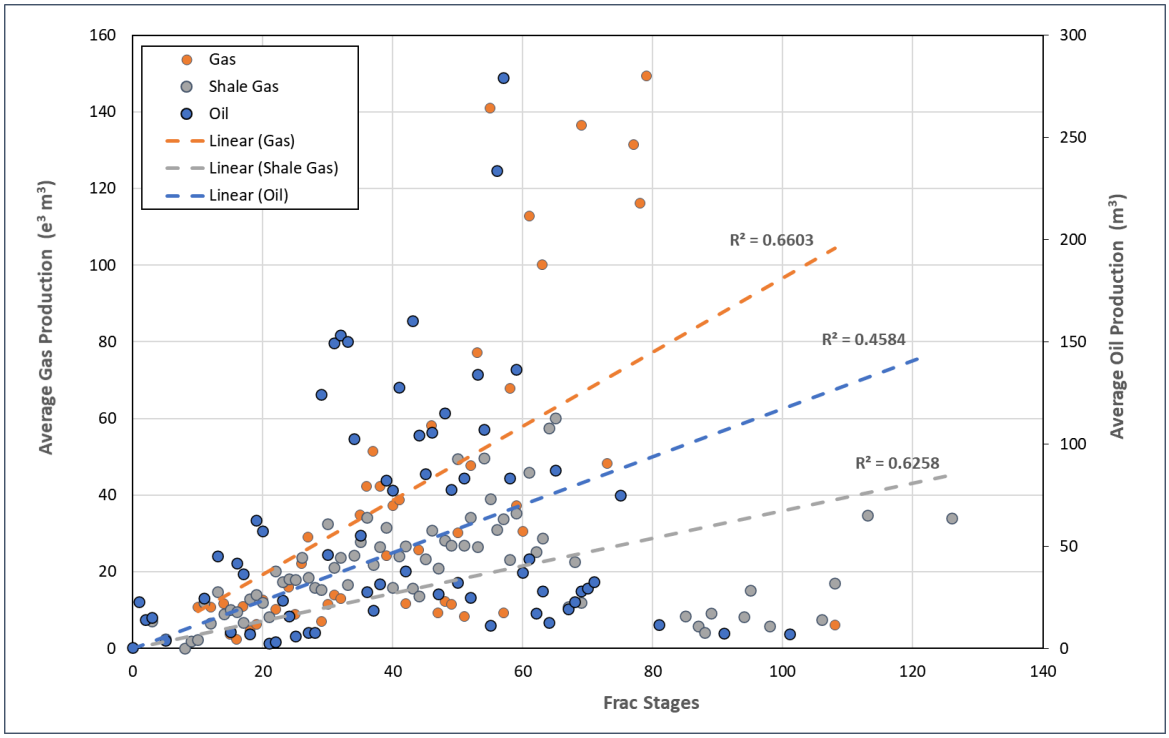


Figure 6-10: Frac stages vs. production rate (data source [60])

6.5 Total Depth (TD)

The Duvernay is a deep play (ranging from 2,029 to 3,960 m in wells completed in Duvernay formation) and deep to the southwest, which adds to the formation drilling costs, and it believes that it is home to the most expensive onshore wells in North America. Well depth statistical information for different well types is presented in Table 6-1. Histogram of well count for different TD has shown in Figure 6-11. Ninety-eight percent of the wells have a depth greater than 4,000(m). The depth of wells has an increasing trend from northeast towards the southwest area of the play.

Table 6-1: Total Drilling Depth (TD) for Duvernay wells (data source [60])

Fluid Type	Min of TD	Max of TD	Average of TD
Gas	2430	7505	5647
Oil	2394	6550	5228
Shale Gas	3905	7770	5754

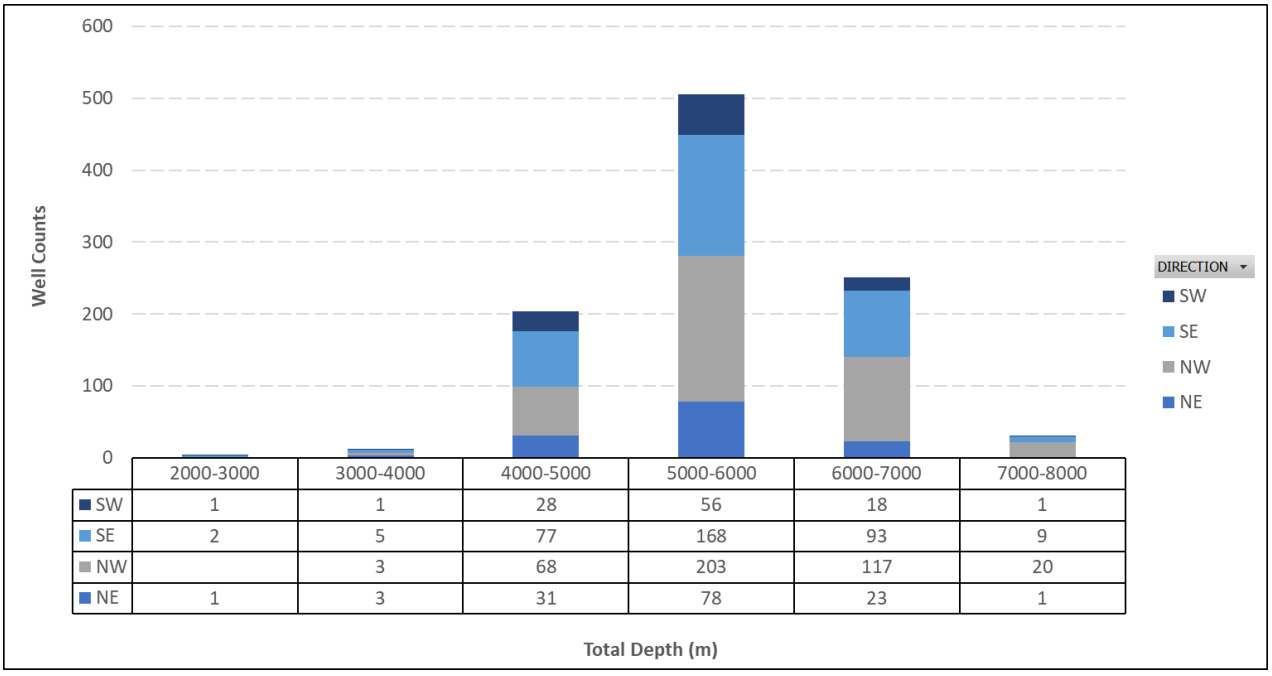


Figure 6-11: Total depth distribution in Duvernay wells (data source [60])

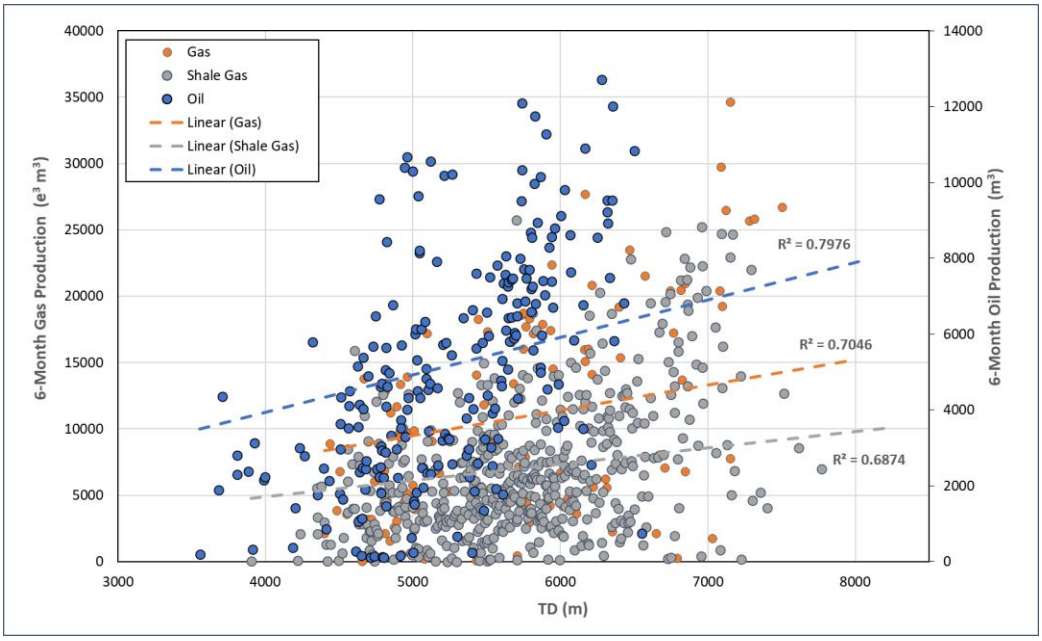


Figure 6-12:Totat depth vs. production rate (data source [60])

6.6 Thickness

The shale members within the Duvernay formation are the primary target zones for unconventional development. Net pay varies considerably across the play, with thickness ranging from 2 to 151 meters in selected wells in the area. The AER subdivides the Duvernay into three informal lithostratigraphic members: A shale, B carbonate, and C shale. The A & C shale members are the target zones for development, while the B carbonate member may restrict the propagation of hydraulic fractures when it exceeds a critical thickness and is considered to be non-reservoir [55].

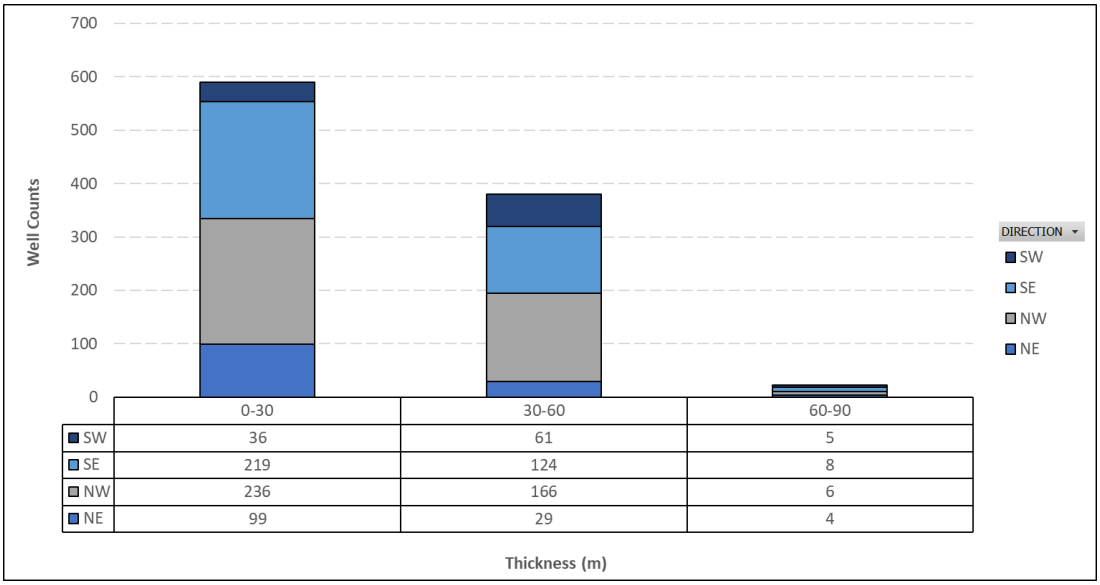


Figure 6-13: Thickness distribution in Duvernay (data source [60])

Most of the acreage in the Duvernay is held by a block of large companies. The Duvernay is liquids-rich, and the condensate it produces can be used as a diluent to help get the bitumen to market. Because of this, condensate prices out of the Duvernay actually have exceeded crude oil prices in the region many times in the recent past. Development in the Duvernay began in 2011 with horizontal multistage fracturing and has steadily Increased.

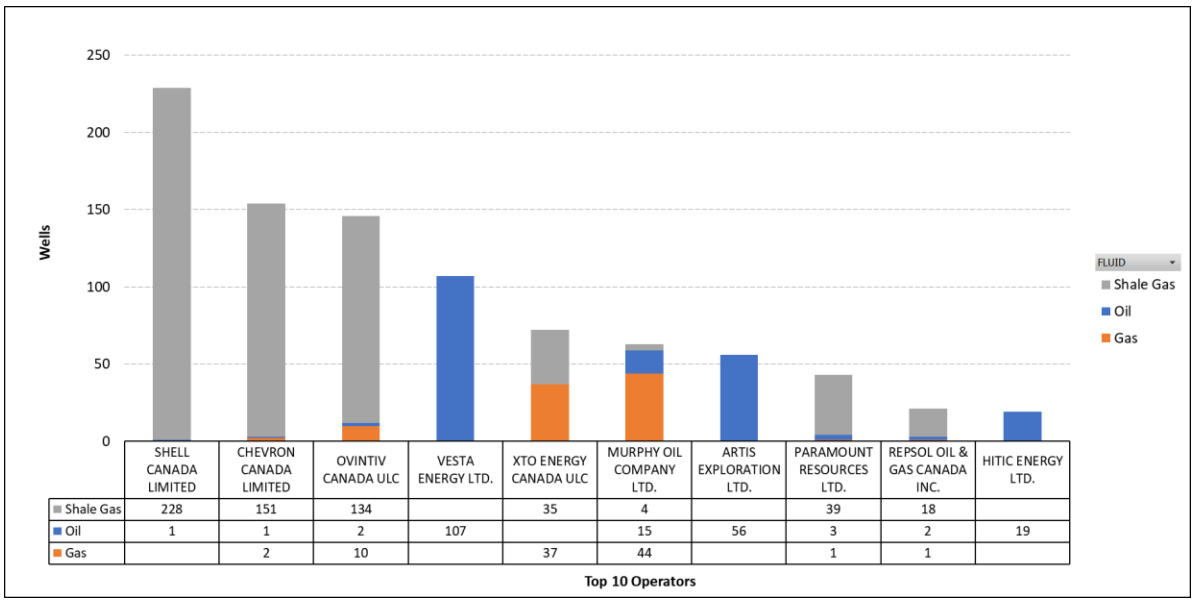


Figure 6-14: Top 10 producers in Duvernay (data source [60])

6.7 Well Orientation

Well orientation defines as the direction in which the horizontal section is laying, from 0-360 deg. More specifically, the number of degrees the bottom hole location is rotated from the surface location, with 0 deg at true north. To determine the effect of well orientation on production, first, we need to calculate the orientation.

For this, for each well, we assume an origin at the surface hole location (approximate heel location) and the endpoint at the bottom hole location (toe). Then connect the dots to draw a line or well stick. In reality, no wells follow this path, but it simplifies the calculation and should not affect the result. Now we can classify the well direction into four groups, NE, SE, SW & NW, with the below statements:

$$NS\ Direction \rightarrow \begin{cases} Bottom\ Hole\ Latitude > Surface\ Latitude \rightarrow\ N \\ Bottom\ Hole\ Latitude < Surface\ Latitude \rightarrow\ S \end{cases} \quad (6\text{-}1)$$

$$EW\ Dirction \rightarrow \begin{cases} Bottom\ Hole\ Longitude > Surface\ Longitude \rightarrow E \\ Bottom\ Hole\ Longitude < Surface\ Longitude \rightarrow W \end{cases} \quad (6\text{-}2)$$

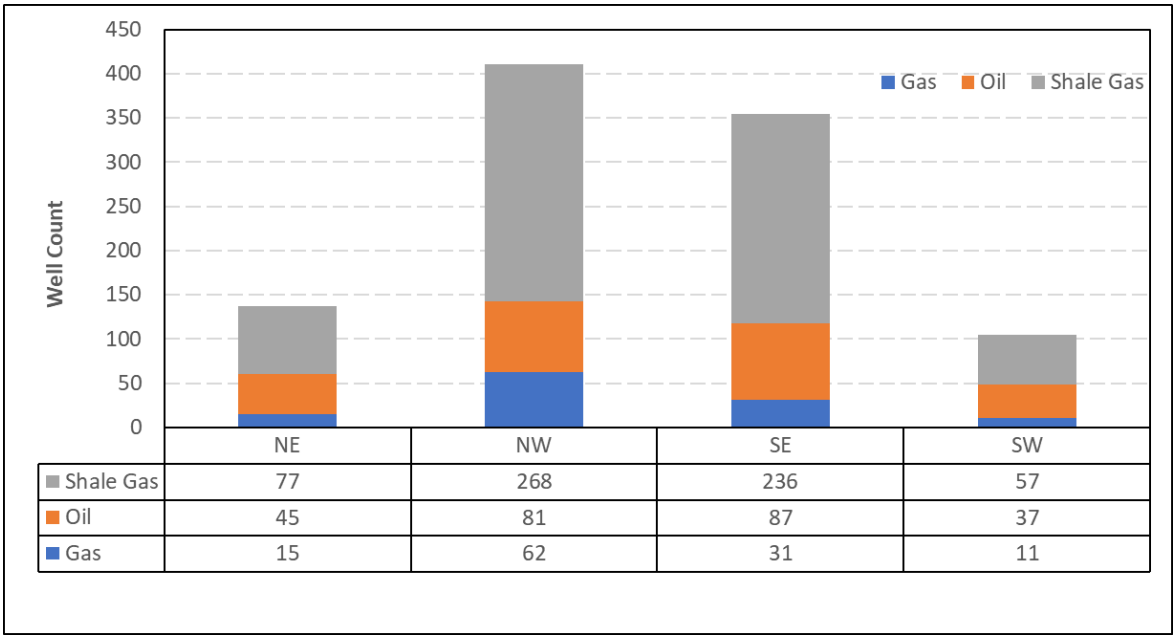
Having a quadrant for each well, we can calculate the azimuth, which is the horizontal angle measured clockwise starting true north (0 degrees). We can also generalize the direction from 0 to 180 degrees and ignore the heel/toe placement. Considering $\boldsymbol{\Delta x}$ as longitude offset and $\boldsymbol{\Delta y}$ as latitude offset, we can calculate orientations as below:

$$\Delta x = Abs(Surface\ Longitude - bottomhole\ Longitude) \quad (6\text{-}3)$$

$$\Delta y = Abs(Surface\ Latitude - bottomhole\ Latitude) \quad (6\text{-}4)$$

$$\begin{cases} A = ArcTan\left(\frac{\Delta x}{\Delta y}\right) \times \left(\frac{180}{\pi}\right) & (NE) \\ A = ArcTan\left(\frac{\Delta y}{\Delta x}\right) \times \left(\frac{180}{\pi}\right) + 90 & (SE) \\ A = ArcTan\left(\frac{\Delta y}{\Delta x}\right) \times \left(\frac{180}{\pi}\right) + 180 & (SW) \\ A = 360 - ArcTan\left(\frac{\Delta x}{\Delta y}\right) \times \left(\frac{180}{\pi}\right) & (NW) \end{cases} \quad (6\text{-}5)$$

Now that each well has been classified, we can summarize the result. Figure 6-15 shows the wells developed in different directions. As it is evident in this figure, most of the wells have NW and SE directions perpendicular to the global stress trend.



	NE	NW	SE	SW
Shale Gas	77	268	236	57
Oil	45	81	87	37
Gas	15	62	31	11

Figure 6-15: Well count per direction (data source [60])

6.8 Stress Map

To characterize stress patterns and sources in the Duvernay area, the crustal present-day stress field data from World Stress Map (WSM) project has been analyzed [61]. In unconventional well planning, the stress field and its orientation relative to well trajectory are important. The orientation shows the maximum horizontal compressional stress (S_{Hmax}) which measured from different

source such as wellbore breakout, drilling induced fractures, and in-situ stress measurements (hydraulic fracturing). The well sticks and the stress orientation have demonstrated in Figure 6-16. As it is clear, the general direction of stress in the area is NE which is perpendicular to the general direction of most of the wells. As expected, most of the wells (76% of the total wells) in the area have drilled in the direction perpendicular to the general stress direction.

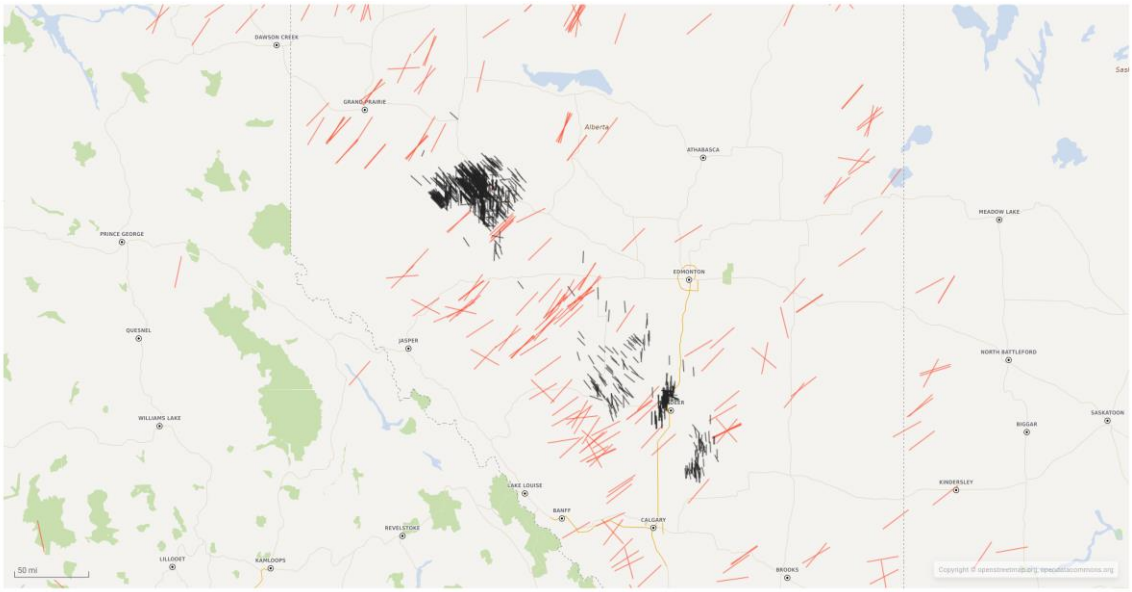


Figure 6-16: Stress map (red lines) and well sticks (black lines) in Duvernay (data source [61])

6.9 Production

To analyze different operators' activity in Duvernay and explore their rank over the monthly span in 2019, their monthly production rates have been compared in bump charts of Figure 6-17 and Figure 6-18. Based on these charts, we can easily compare the performance of different operators. Vesta Energy, Artis Exploration, and Murphy Oil were top oil producers over the entire 2019 months; however, Shell, Chevron, XTO, and Ovintiv were the main competitors for gas production. Starting in 2019, Shell was the top gas producer in January, and then Shell got the place, and this trend changed by Chevron and XTO for the reset of 2019.

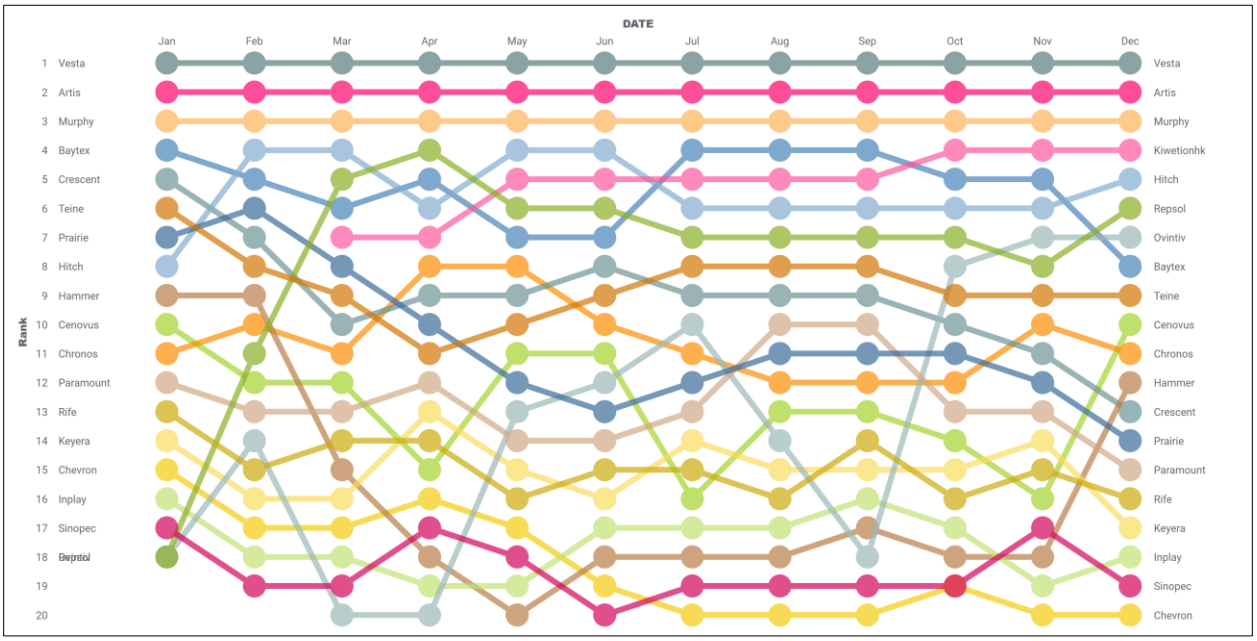


Figure 6-17: Operators' monthly oil production ranking in 2019 (data source [60])

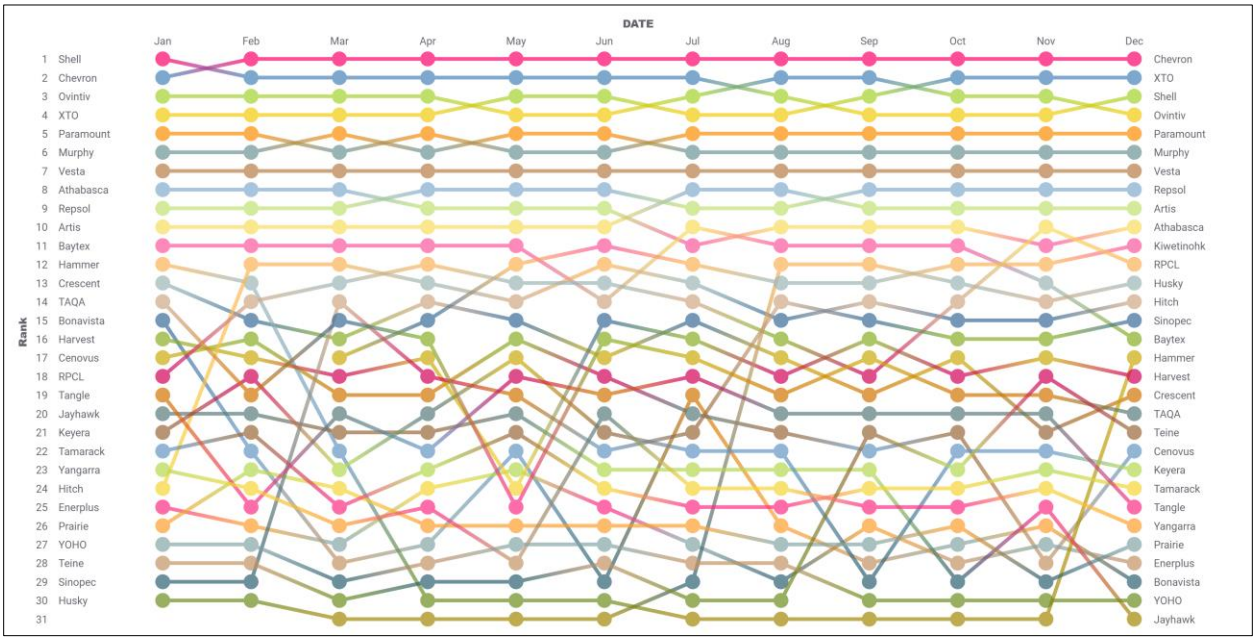


Figure 6-18: Operators' monthly gas production ranking in 2019 (data source [60])

6.10 Oil & Gas Price

In the past decade, the oil and gas price had many up and down, and this behavior directly impacted the production rate in the Duvernay area. Figure 6-19 and Figure 6-20 show the oil and gas price since 2010 and its impact on production. A drastic decline in oil price and consequently in its production happened in early 2020 due to pandemic, while it has not affected the gas market.

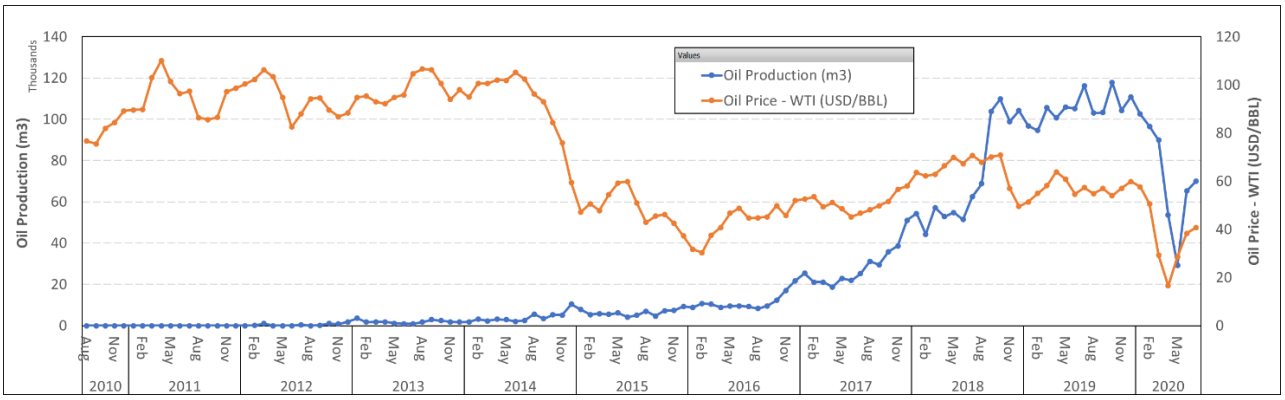


Figure 6-19: Average oil production and its variation vs. time and oil price (data source [60], [62])

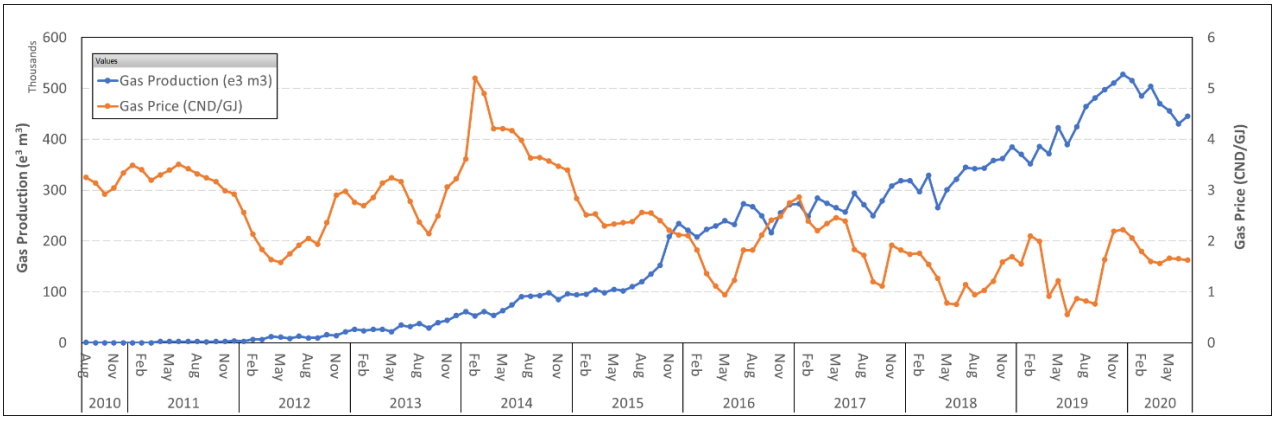


Figure 6-20: Average monthly gas production and its variation vs. time and gas price (data source [60], [62])

6.11 Seasonal Change

Figure 6-21 depicts the annual change in oil and gas production. Figure 6-22 shows the seasonal change of average oil and gas production rate during different months since production has started. As it shows, July is the most productive month, and September is the less productive

month for gas, but it is different for oil production. December has the highest rate, and September has the lowest rate.

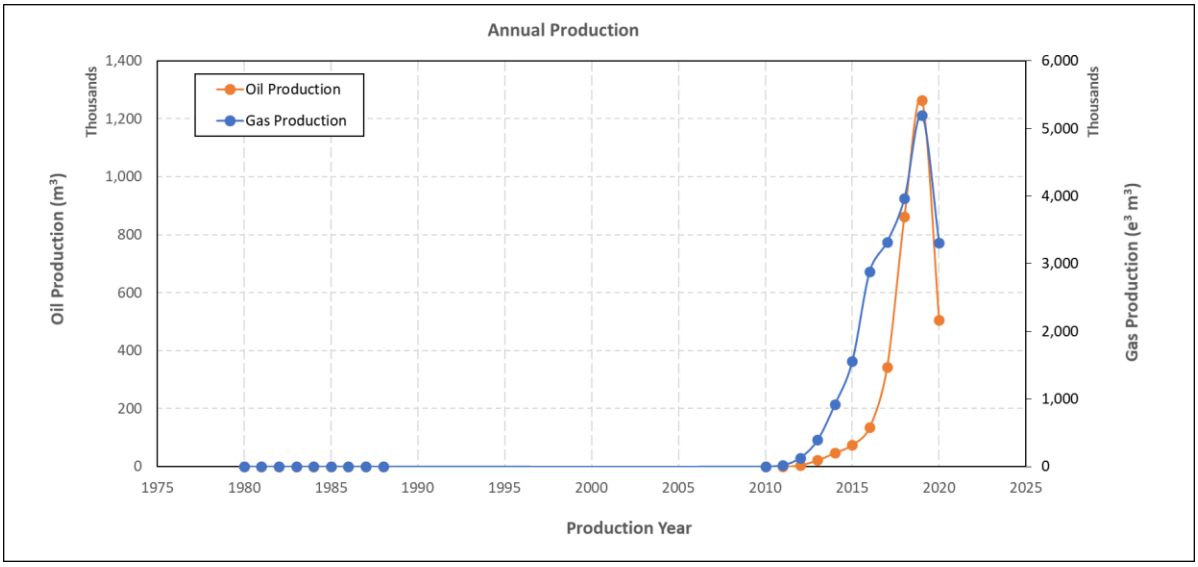


Figure 6-21: Annual production of Duvernay wells (data source [60])

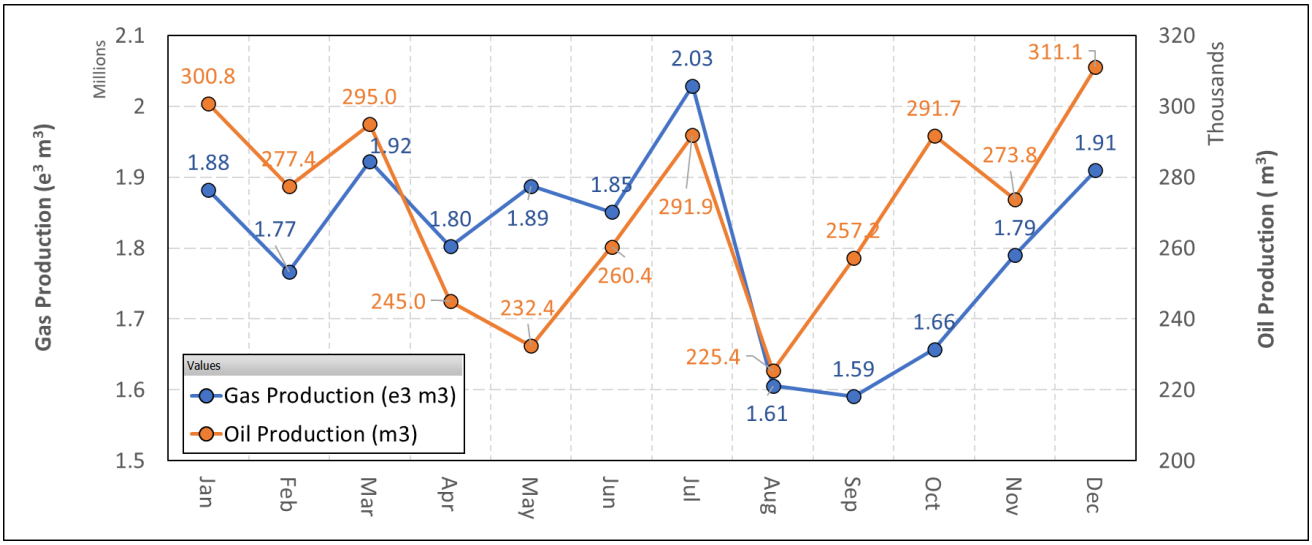


Figure 6-22: Seasonal cumulative oil and gas production in Duvernay (data source [60])

7 DEVELOP FORECASTING MODULES FOR DUVERNAY PLAY

In this section, we will develop different modules to forecast the production rate in Duvernay wells. First, the forecast will be performed by the traditional approach, i.e., decline curves for all wells, and then different machine learning algorithms will be examined to find the most appropriate one. Finally, these results will be compared with time series forecasting.

7.1 Analytical Module

Different DCA algorithms have been applied to the wells with enough production history (at least with one-year history) and then one with the best fit selected to calculate Estimated Ultimate Recovery (EUR). Figure 7-1 to Figure 7-3 shows the DCA curves for three wells with the highest history data.

Then we will expand our scale to field scale and analyze the DCA curve for the most productive fields: Kaybob South and Willesden Green. As shown in Figure 7-4, the Kaybob-South field has a growing trend up to 2020, and traditional models fail to explain this behavior. This behavior is highly tied to the number of active wells in production (Figure 7-5), which itself related to the gas price in the market. If we divide the production data into 2 phases, there is not enough data to find a reasonable decline curve in declined part (orange markers), which will overestimate the EUR value. Figure 7-6 shows the production history in the Willesden-Green field. It shows growth in production rate up to 2014 due to field development activities and drilling more wells. Then field

starts to decline in production rate, and there is enough production history data to fit a decline curve.

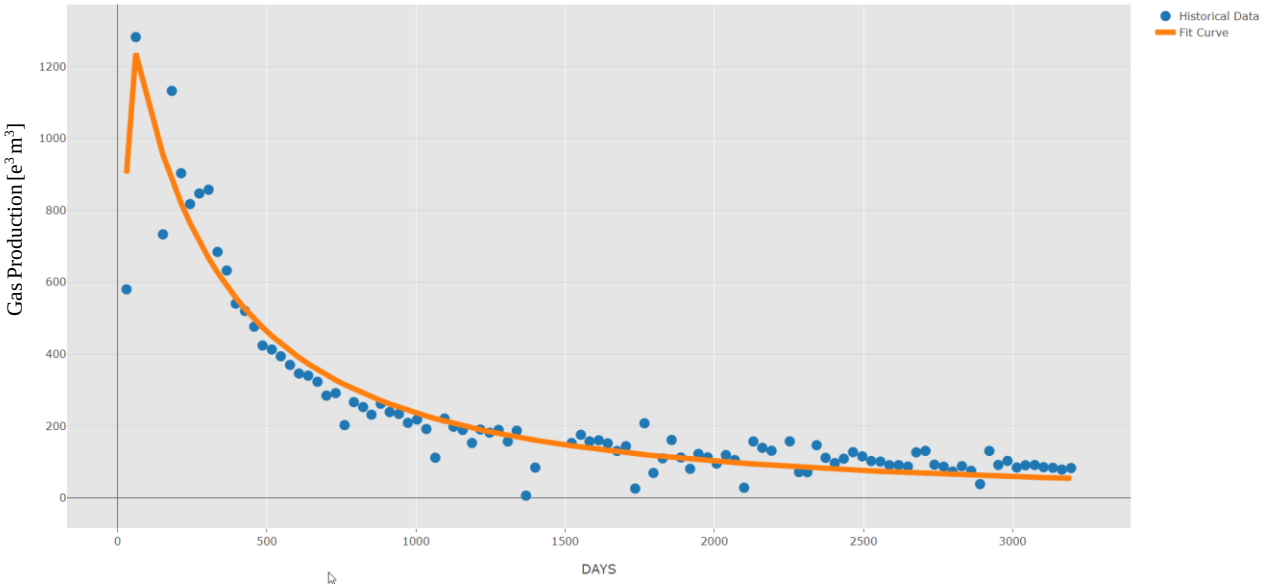


Figure 7-1: Decline curve for well 100/07-16-044-07W5/2 (Willesden Green)

Fit data: $q_i = 1510, D_i = 0.0035, b = 0.62, EUR = 947{,}445(e3\ m3), R^2 = 96\%$

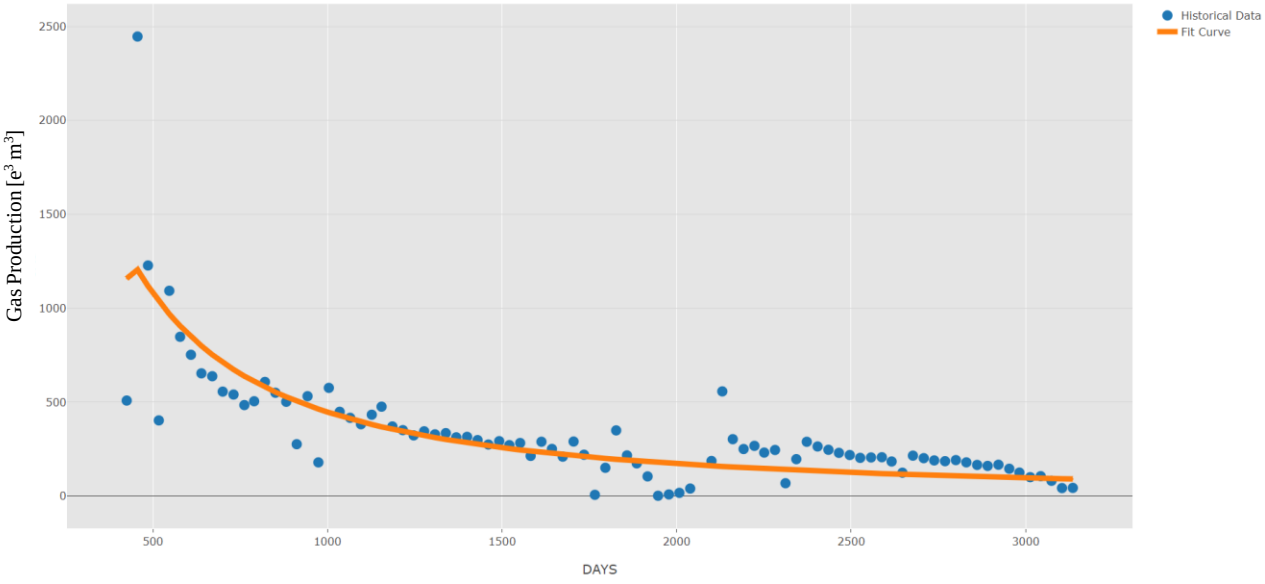


Figure 7-2: Decline curve for well 102/13-22-062-21W5/0 (Kaybob South)

Fit data: $q_i = 12231.5, D_i = 0.0121, b = 0.67, EUR = 1{,}508{,}940(e^3\ m^3), R^2 = 80\%$

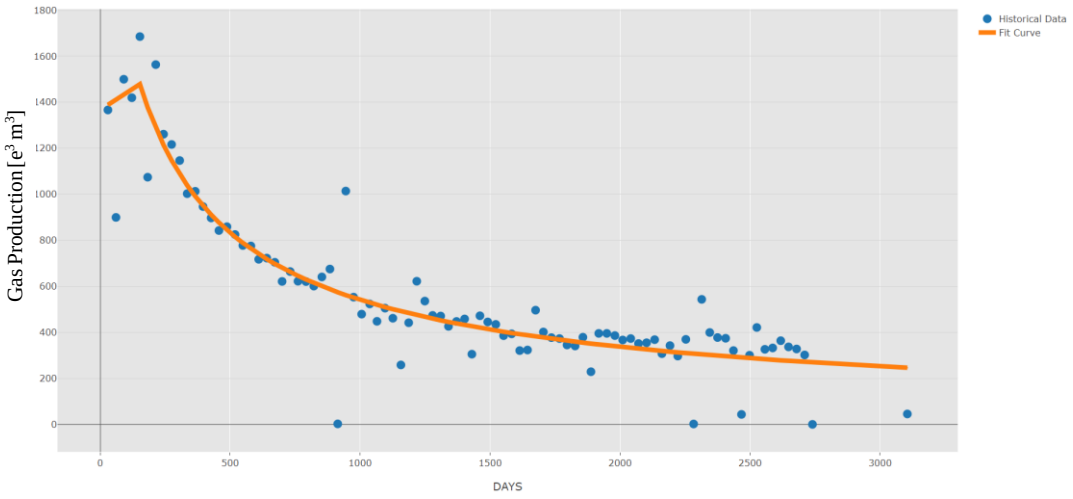


Figure 7-3: Decline curve for well 100/03-13-060-20W5/0 (Kaybob South)

Fit data: $q_i = 2495.8, D_i = 0.0049, b = 1.31, EUR = 7{,}547{,}075 (e^3\ m^3), R^2 = 92\%$

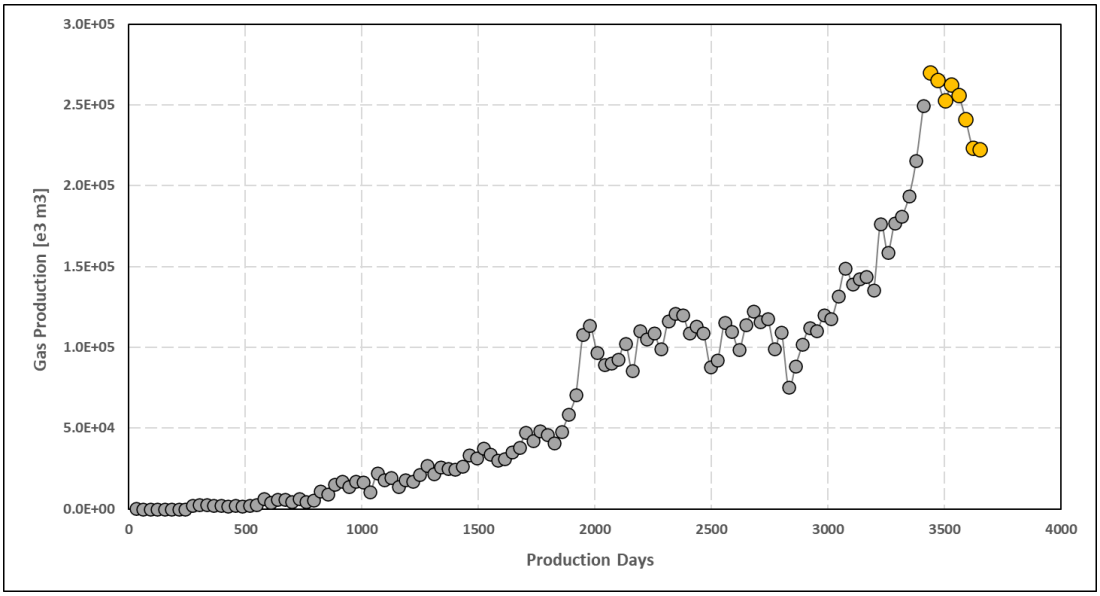


Figure 7-4: Production curve for Kaybob South field

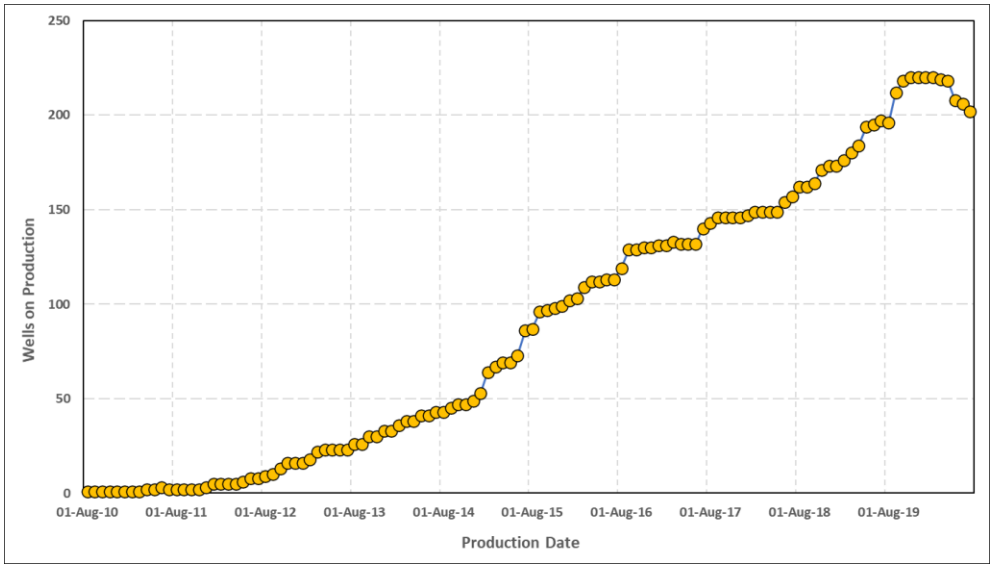


Figure 7-5: Active production wells vs. time for Kaybob South field

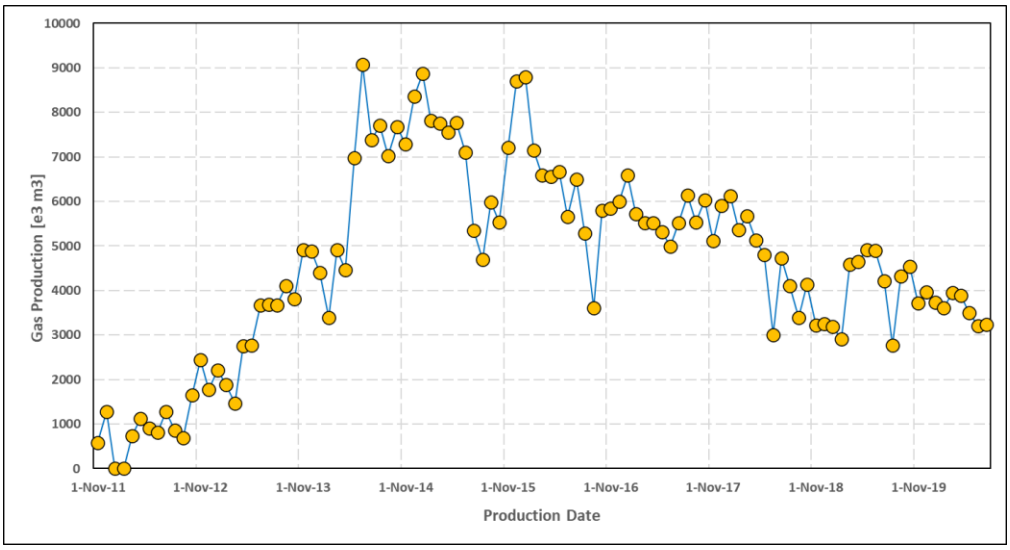


Figure 7-6: Production curve for Willesden Green field

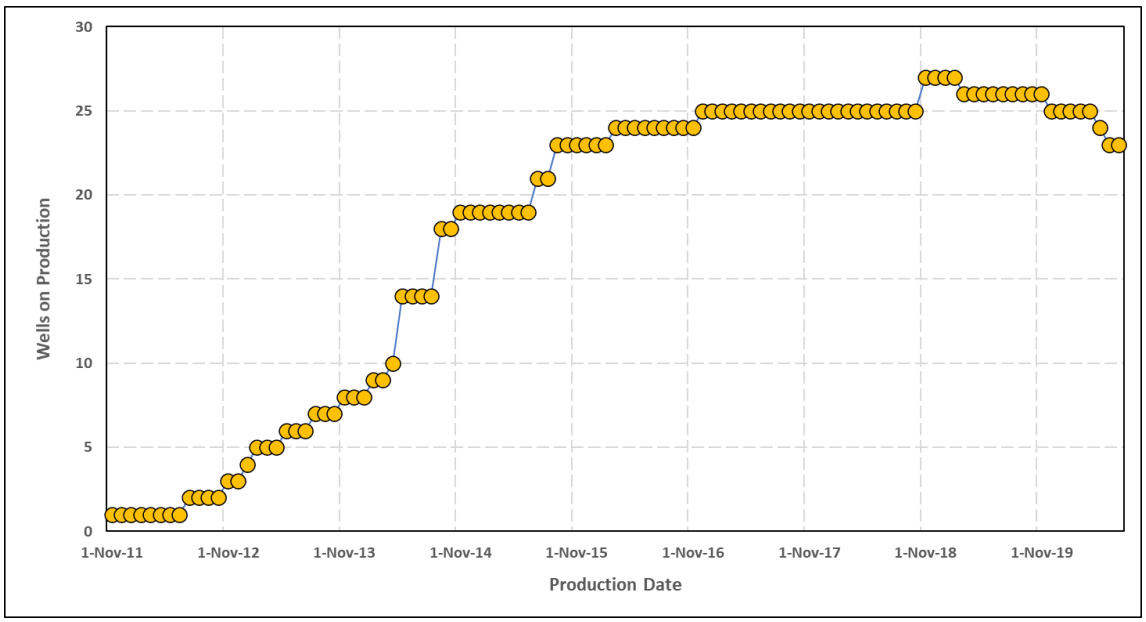


Figure 7-7: Active production wells vs. time for Willesden-Green filed

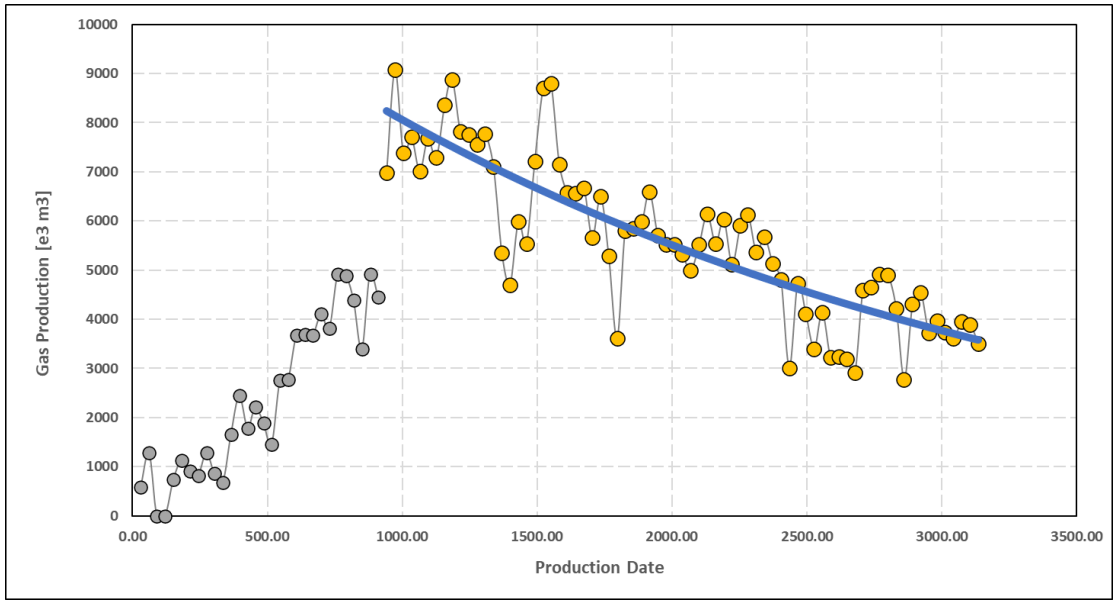


Figure 7-8: Decline curve for WILLESDEN-GREEN field

Fit data: $q_i = 11769, D_i = 0.00038, b = 1, EUR = 31{,}460{,}926 (e^3\ m^3), R^2 = 71\%$

7.2 Machine Learning

7.2.1 Initial Production Estimation

To build a machine learning model to estimate the initial production rate in Duvernay well, a dataset from a different database, including well productions, well header data, well completion, and frac has been collected and to evaluate the well performance, cumulative productivity of gas during the early phase of production within six months has calculated and considered as target feature for ML models. After preprocessing, different ML models have been created: Linear Regression, Gradient Boosting, Random Forest, Neural Network, and AdaBoost. To evaluate the performance of different algorithms, 20% of samples were randomly selected for testing, and the remaining sample was used to train the models.

In Figure 7-9, feature importance has used to explain the input features. Figure 7-10 explains which features contribute the most and how they contribute toward the prediction. Features in the plot are ordered by their relevance to the prediction. TVD is the most important feature to predict initial gas production.

The result of cross-validation with 10-fold samples has listed in Table 3-1. AdaBoost has the highest ($R^2 = 76\%$) and Linear Regression ($R^2 = 53\%$) the lowest accuracy. To aggregate different models and improve the prediction, all models are stacked with an ensembling method that computes a meta-model from several base models. In terms of aggregation of input models, Ridge Regression has been implemented. The stacking could improve the prediction performance by 1%. Figure 7-11 shows the correlation between different ML models.

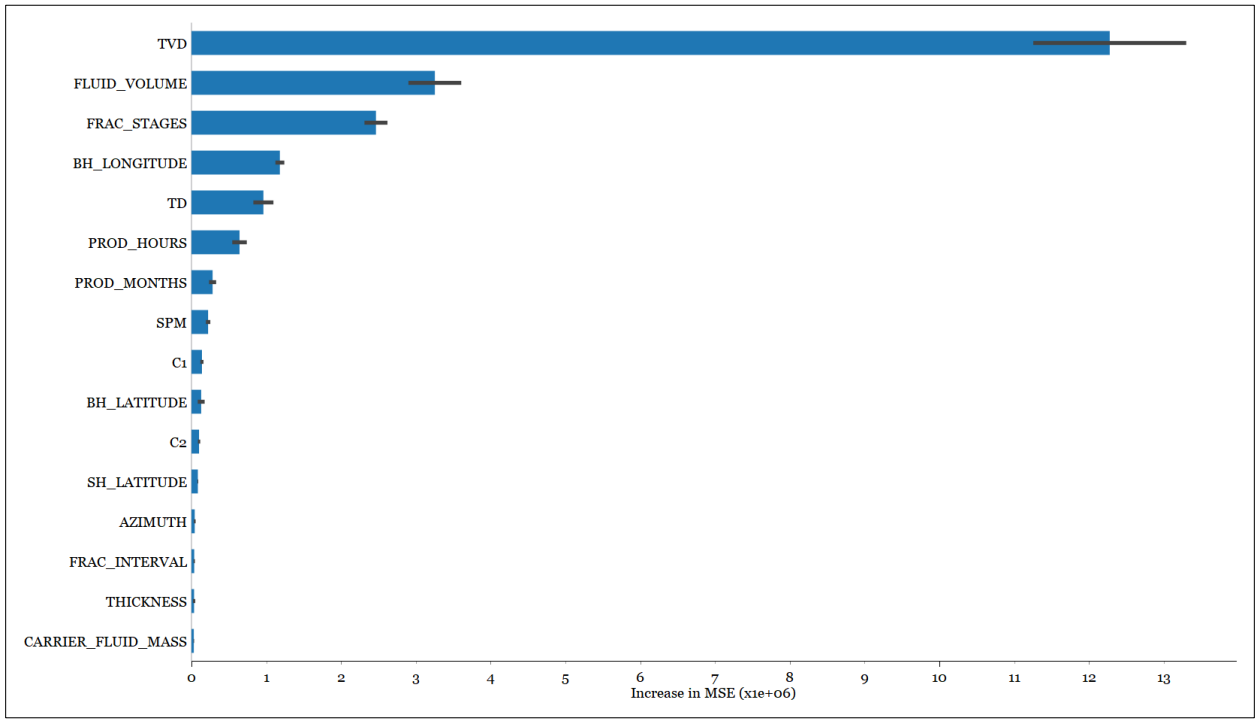


Figure 7-9: Feature importance based on permutation technique

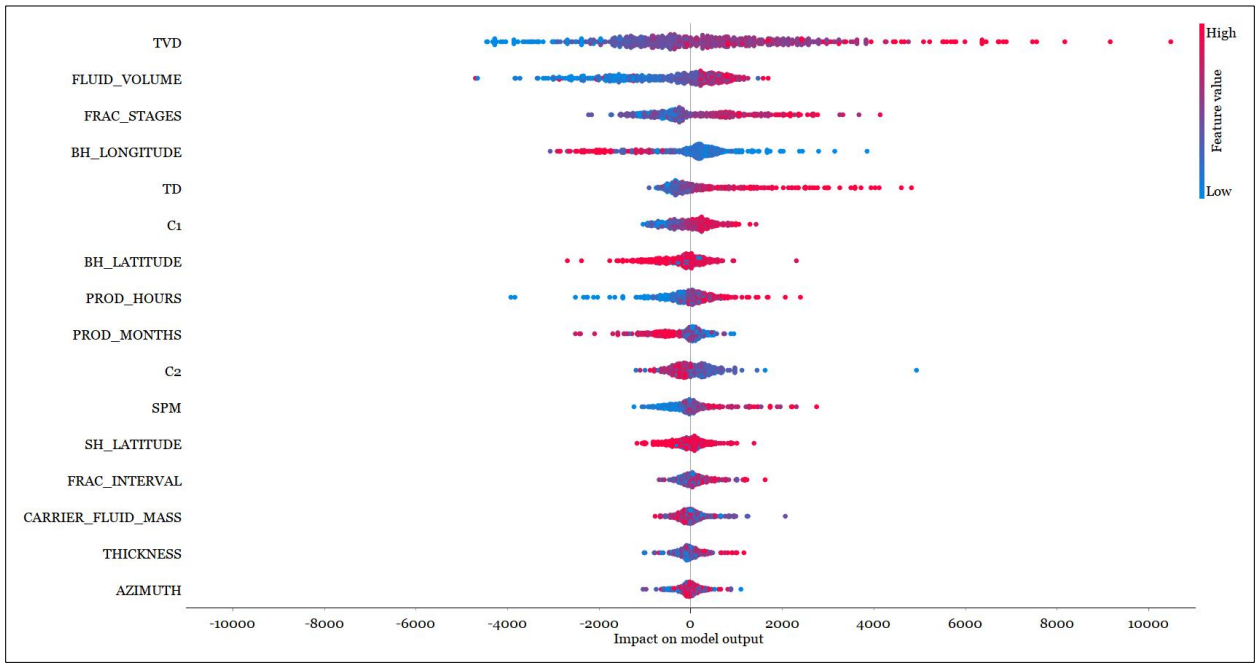


Figure 7-10: Explain the impact of input features on model output

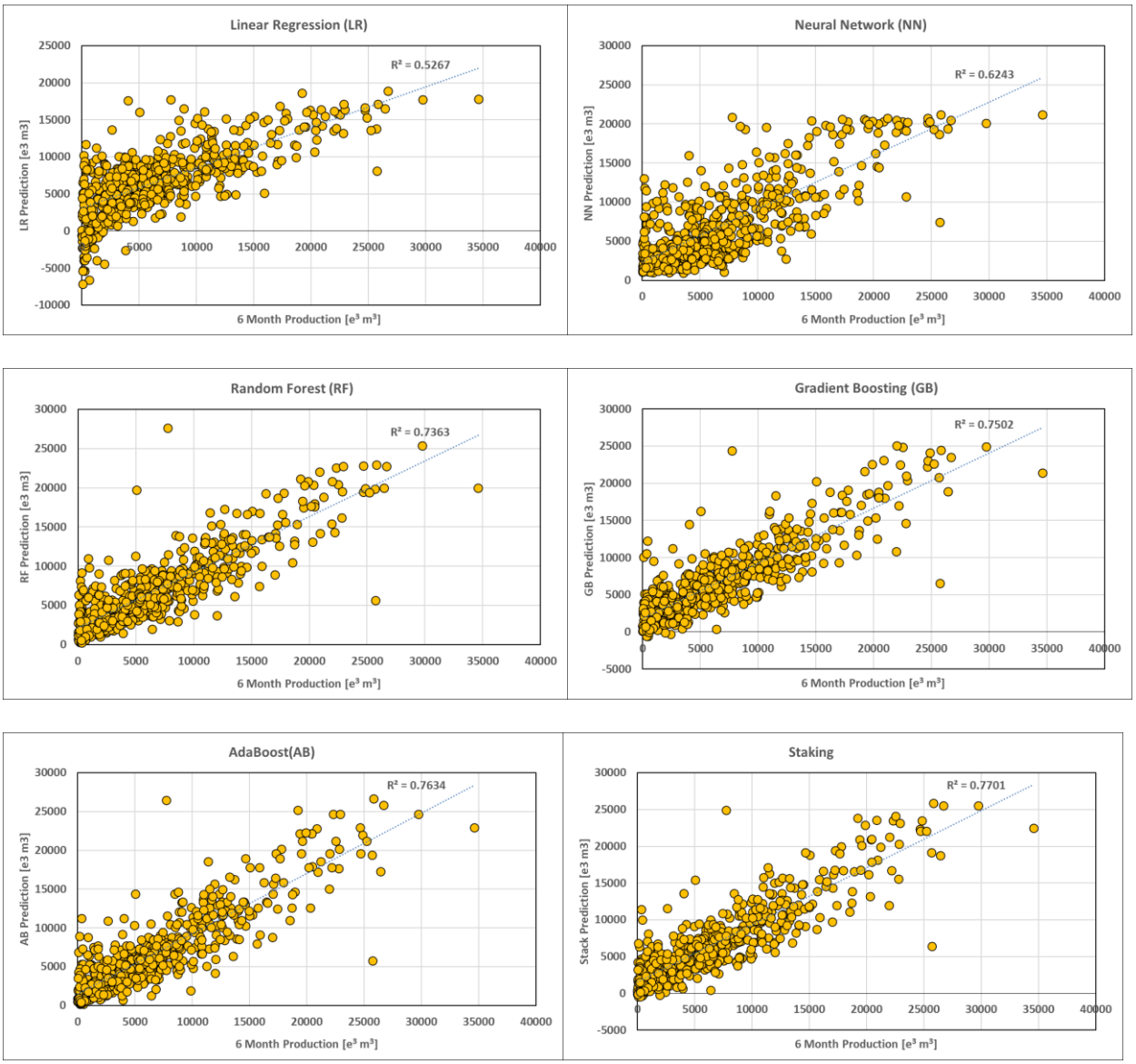


Figure 7-11: correlation plot between predicted data and 6-month gas production data

Table 7-1: Evaluation result of ML prediction for 6-month production

Model	**MSE**	**RMSE**	**MAE**	**R^2**
Linear Regression	16135435	4017	3065	0.53
Neural Network	12991445	3604	2604	0.62
Random Forest	9025974	3004	2018	0.73
Gradient Boosting	8509823	2917	2037	0.74
AdaBoost	8060718	2839	1886	0.76
Stack	7825960	2797	1902	0.77

7.3 Time Series Forecasting

Figure 7-12 represents the workflow to build the forecasting model based on the time series algorithm. Monthly gas production, gas price, production hours, and production date have been selected from the database and then transformed as a time series dataset. Two different models have been built based on ***ARIMA*** (Auto-Regressive Integrated Moving Average) and ***VAR*** (Vector Autoregression) algorithms.

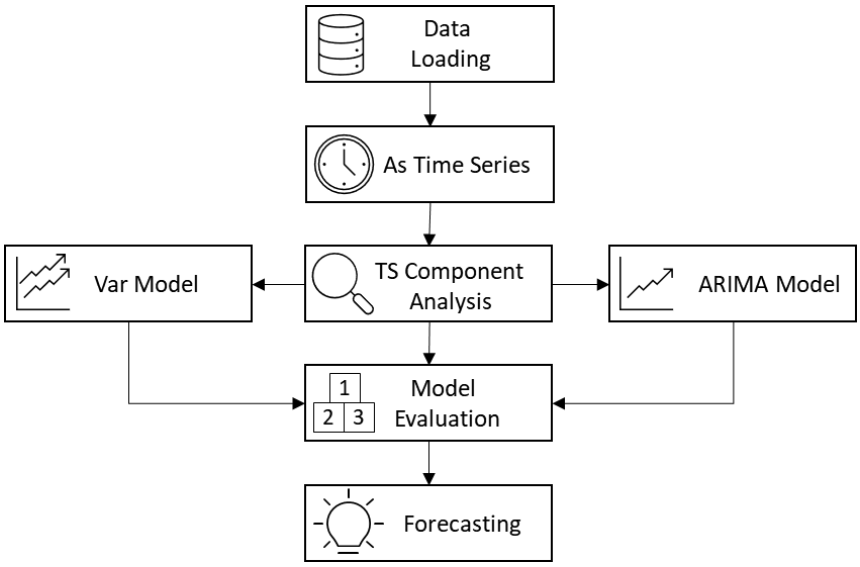


Figure 7-12: Workflow for time series modeling

Figure 7-13 shows the variation of the monthly gas production time series. As it is clear, gas production has an increasing trend up to 2020 and then decreased mainly due to the pandemic. The time plot shows some sudden changes, mainly two jumps in 2016 and 2019 and a drop in 2020.

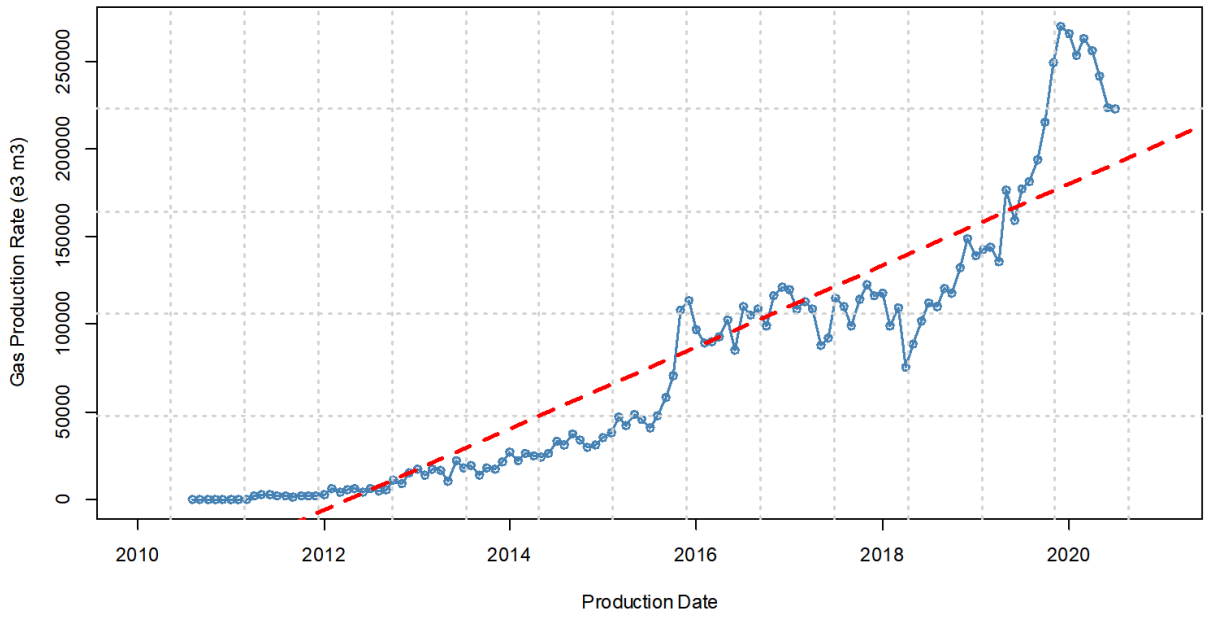


Figure 7-13: Time series change of monthly gas production in Kaybob South field

Figure 7-14 shows the seasonal effect on gas production rate. The variance and the mean in monthly cyclic are high in July and are low in August.

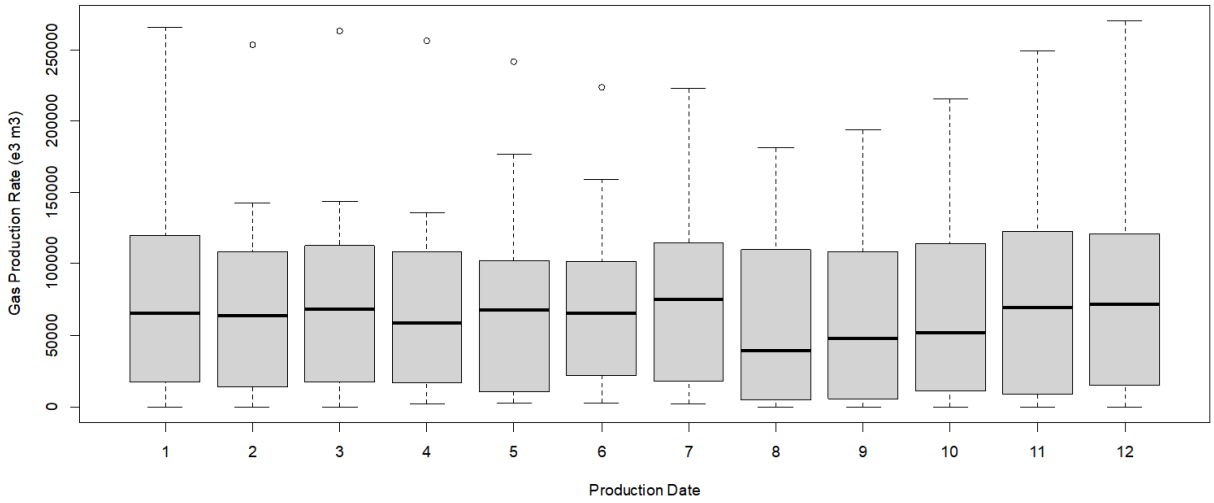


Figure 7-14: Cyclic changes of gas production time series in Kaybob South

7.3.1 Decomposition

Plotting time series data is an important first step in analyzing their various components. Beyond that, however, we need a more formal means for identifying and removing characteristics such as

a trend or seasonal variation. The decomposition model reduces a time series into three components: trend, seasonal effects, and random errors. This is often

done to help to understand the time series, but it can also improve forecast accuracy [63]. If we assume an additive decomposition, then we can write:

$$y_t = S_t + T_t + R_t \tag{7-1}$$

Where $\boldsymbol{S_t}$ is the seasonal component, $\boldsymbol{T_t}$ is the trend-cycle component, $\boldsymbol{R_t}$ and is the remainder component, all at period. Alternatively, a multiplicative decomposition would be written as:

$$y_t = S_t \times T_t \times R_t \tag{7-2}$$

The additive decomposition is the most appropriate if the magnitude of the seasonal fluctuations, or the variation around the trend cycle, does not vary with the time series level.

Figure 7-15 shows an additive decomposition of time series data. The three components are shown separately in the bottom three panels. These components can be added together to reconstruct the data shown in the top panel. Notice that the seasonal component changes twice annually and is the same pattern in any consecutive years have similar patterns. The remainder component shown in the bottom panel is what is left over when the seasonal and trend-cycle components have been subtracted from the data.

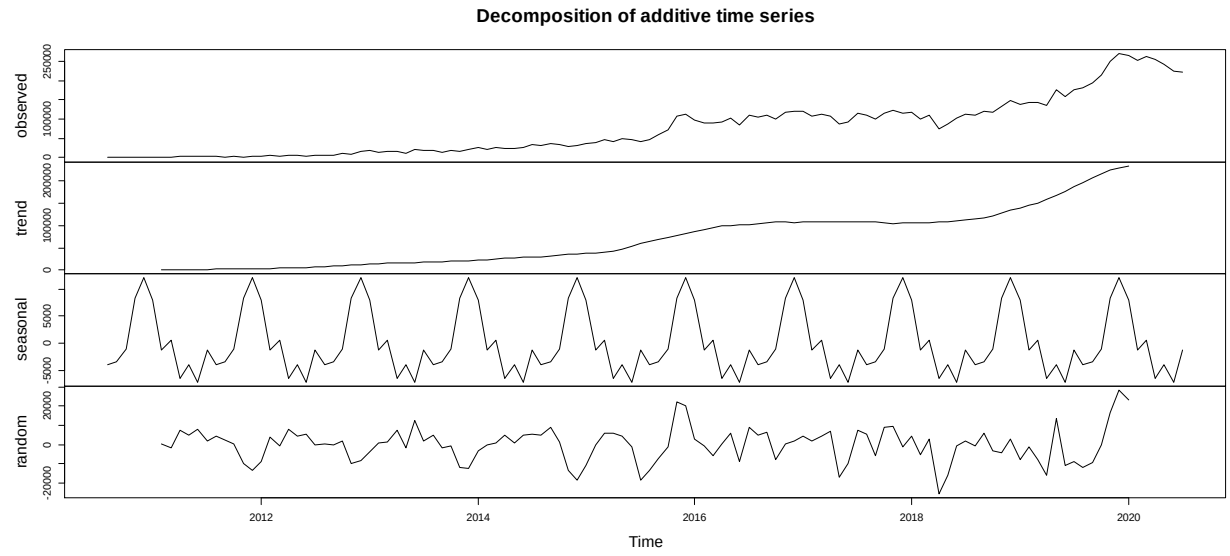


Figure 7-15: Time series decomposition for Kaybob South

To see what the trend-cycle estimate looks like, we plot it along with the original data in Figure 7-16. Notice that the trend cycle (in red) is smoother than the original data and captures the main movement of the time series without all of the minor fluctuations.

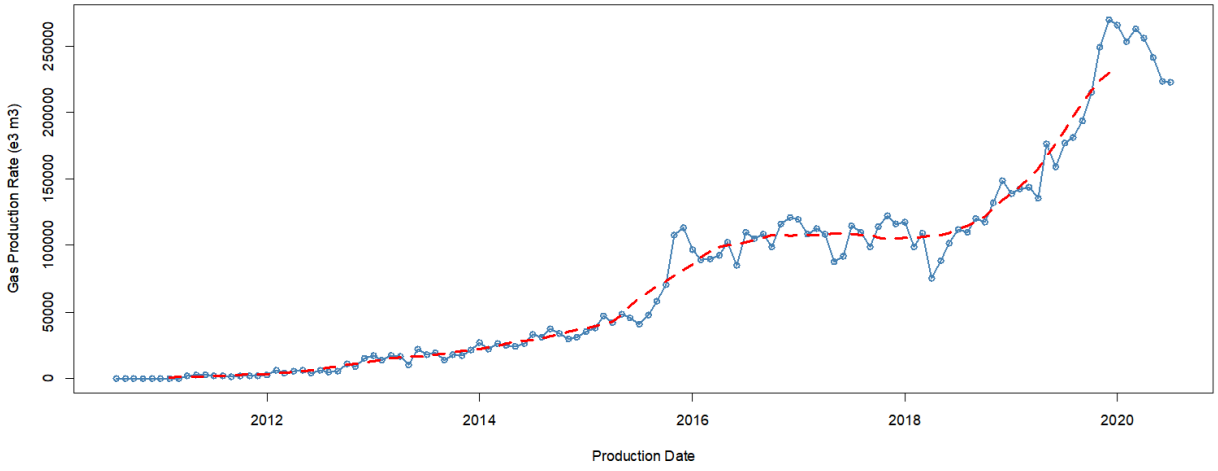


Figure 7-16: Gas production trend (red) along with the original data (blue) in Kaybob South

7.3.2 Stationary

A stationary time series is one whose properties do not depend on the time at which the series is observed. In general, a stationary time series will have no predictable patterns in the long term [54]. Augmented Dickey-Fuller (***ADF***) [63] test can reveal stationarity in the time series:

$$ADF = \frac{\hat{\rho}}{S.E(\hat{\rho})} \tag{7-3}$$

where $\hat{\boldsymbol{\rho}}$ is the coefficient estimation and $S.\boldsymbol{E}(\hat{\boldsymbol{\rho}})$ is its corresponding estimation of standard error for each type of linear model. Stationary testing and converting a series into a stationary series are the most critical processes in time series modeling. In cases when we have no-stationary time series, we first need to make it stationary. Here are three commonly used techniques to make a time series stationary [64]:

1- Detrending: to remove the trend component from the time series
2- Differencing: to model the differences of the terms and not the actual term
3- Seasonality: to remove the seasonality component from the time series

As shown in Figure 7-13, gas production is not stationary, and we first need to stationarize the series. ***ADF*** test on gas production time series results:

$$lag\ order = 1, ADF = 1.5674, p\ value = 0.9699$$

Which rejects that time series is stationary. Here is the result after first order differencing:

$$lag\ order = 1, ADF = -6.8809, p\ value = 0.01$$

This shows after differencing, and the time-series change to stationary. Figure 7-17 shows the time series plot after first order differencing. Now we can see the time series is more stationary.

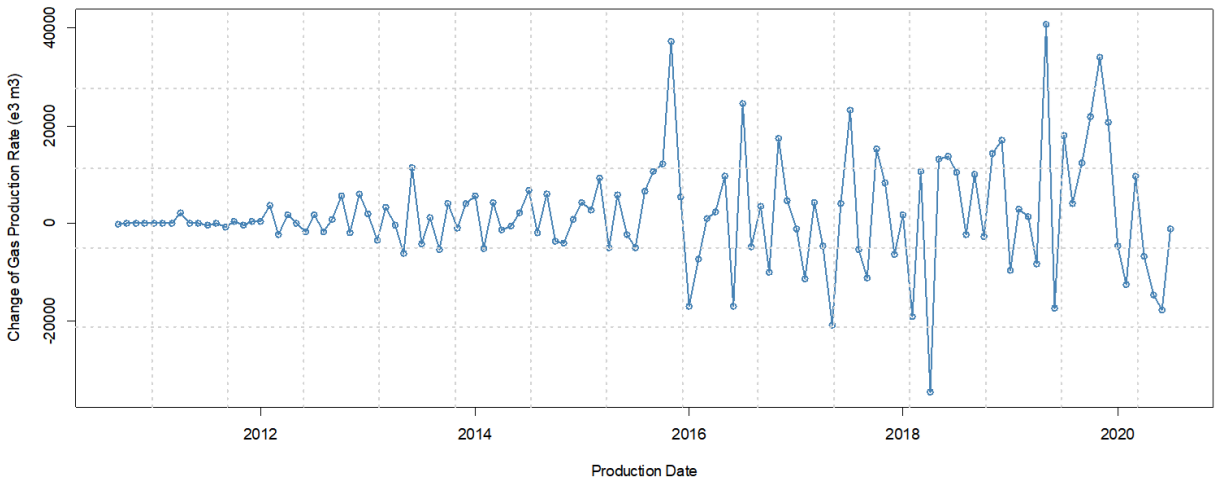


Figure 7-17: time series plot for gas production after the first-order stationary (Kaybob South)

7.3.3 Autocorrelation

An autocorrelation is an important tool for revealing the interrelationships within a time series. It is a collection of correlations, $\boldsymbol{\rho_k}$ for $\boldsymbol{k} = 1, 2, 3, \ldots$ where $\boldsymbol{\rho_k}$ is the correlation between all pairs of data points that are exactly $\boldsymbol{k}$ steps apart.

Visualizing the autocorrelations is much more helpful than listing them. For this purpose, two types of functions are usually applied on time series data: ***ACF*** (Autocorrelation Function) and ***PACF*** (Partial Autocorrelation Function). ***ACF*** measures the relationship between $\boldsymbol{y_t}$ and $\boldsymbol{y_{t-k}}$ for different values of $\boldsymbol{k}$. If $\boldsymbol{y_t}$ and $\boldsymbol{y_{t-1}}$ are correlated, then $\boldsymbol{y_{t-1}}$ and $\boldsymbol{y_{t-2}}$ are also correlated. Then $\boldsymbol{y_t}$ and $\boldsymbol{y_{t-2}}$ might be correlated as they are both connected to $\boldsymbol{y_{t-1}}$. To overcome this problem, we can use ***PACF***, which measure the relationship between $\boldsymbol{y_t}$ and $\boldsymbol{y_{t-k}}$ after removing the effects of lags [54].

The presence of autocorrelations indicates that an ***ARIMA*** model could model the time series and also can help determine the value of $\boldsymbol{p}$ or $\boldsymbol{q}$ of $\boldsymbol{ARIMA(p, d, q)}$ model. We can count the number of significant autocorrelations from the ***ACF***, which is a useful estimate of the number of moving

average (***MA***) coefficients in the model (***q***). PACF helps to identify the number of autoregression (***AR***) coefficients in an ***ARIMA*** model (***p***).

Figure 7-18 shows the autocorrelation functions for gas production time series, one with autocorrelations and one without. The dashed line delimits the significant and insignificant correlations: values above the line are significant. Here from ***PACF***, we estimate that the ***ARIMA*** model will require one AR coefficient ($\boldsymbol{p = 1}$) and from ***ACF*** we can estimate ***MA*** coefficient of one ($\boldsymbol{q = 1}$). That estimate is just a starting point, however, and must be verified by fitting and diagnosing the model.

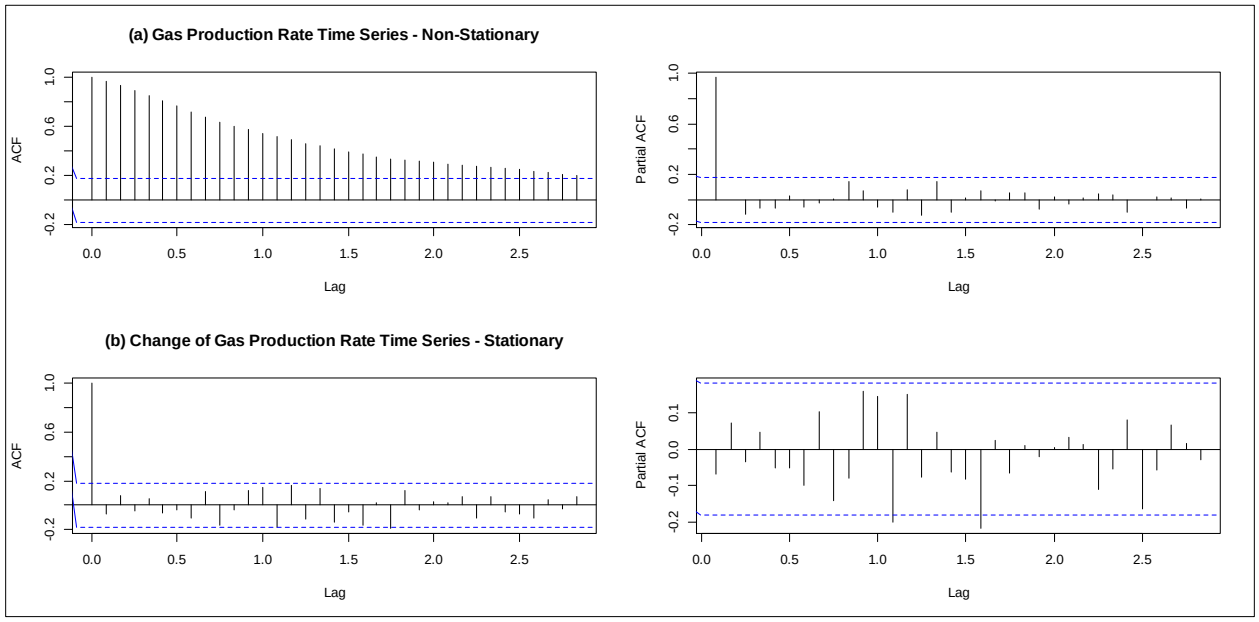


Figure 7-18: ACF and PACF plots before and after differencing (Kaybob South)

7.3.4 ARIMA Forecasting Model

Once the model order has been identified (i.e., the values of ***p*** and ***q***), we need to find the parameters' values that maximize the probability of obtaining the data that we have observed [63]. ***AIC*** is one parameter to select appropriate ***p*** and ***q*** values. Akaike's Information Criterion (***AIC***) can be written as:

$$AIC = -2\log(L) + 2(p + q + k + 1) \quad (7\text{-}4)$$

Where ***L*** is the likelihood of the data, $k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$. For ***ARIMA*** models, the corrected ***AIC*** can be written as [63]:

$$AICc = AIC + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2} \quad (7\text{-}5)$$

T is the total periods in the time series. The Bayesian Information Criterion (***BIC***) can be written as:

$$BIC = AIC + [\log(T) - 2](p + q + k + 1) \quad (7\text{-}6)$$

Good models are obtained by minimizing the ***AIC***, ***AICc***, or ***BIC***. The preference is to use the ***AICc*** [63].

From the previous section, we estimate ***ARIMA*** (1,1,1) as an initial model and then fit different models, calculate ***AICc*** and select the smaller one as the best fit. After running different ***ARIMA*** models and checking the residuals, the model with the order of (0,1,11) and seasonality of (1,0,1) has been selected. Gas production forecasts from the model for the next year are shown in Figure 7-19. 80% and 95% prediction intervals are shown in the figure. Figure 7-20 shows the residuals from the ***ARIMA*** model. All the spikes are within the significance limits, the residuals have no remaining autocorrelations, so the residuals appear to be white noise. Thus, we now have a seasonal ***ARIMA*** model that passes the required checks and is ready for forecasting. Figure 7-21 and Figure 7-22 show the modeling results for Willesden Green filed.

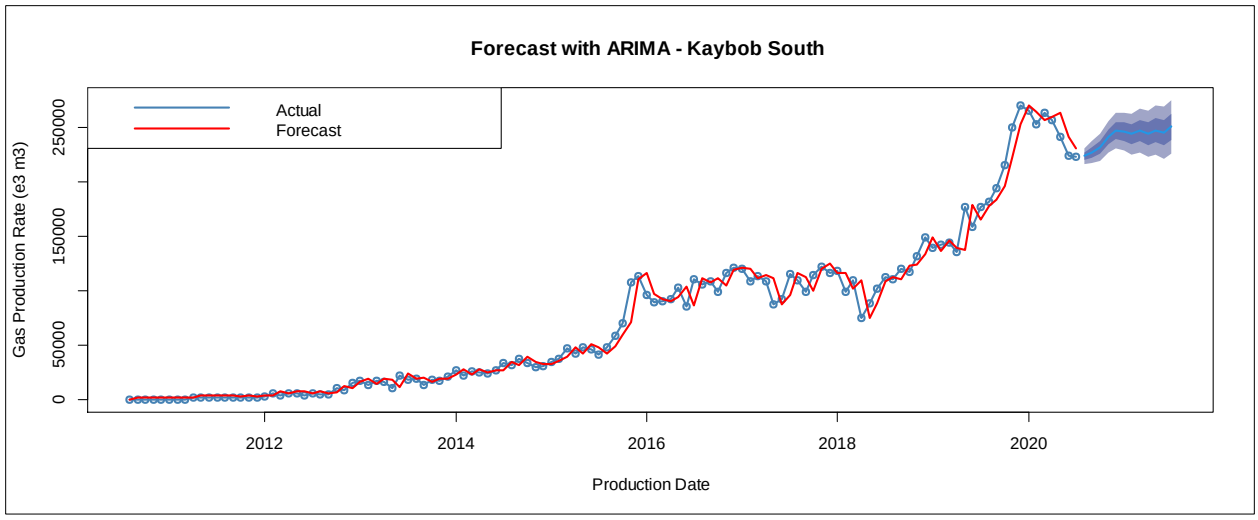


Figure 7-19: Gas production forecasts of Kaybob South field with ARIMA model

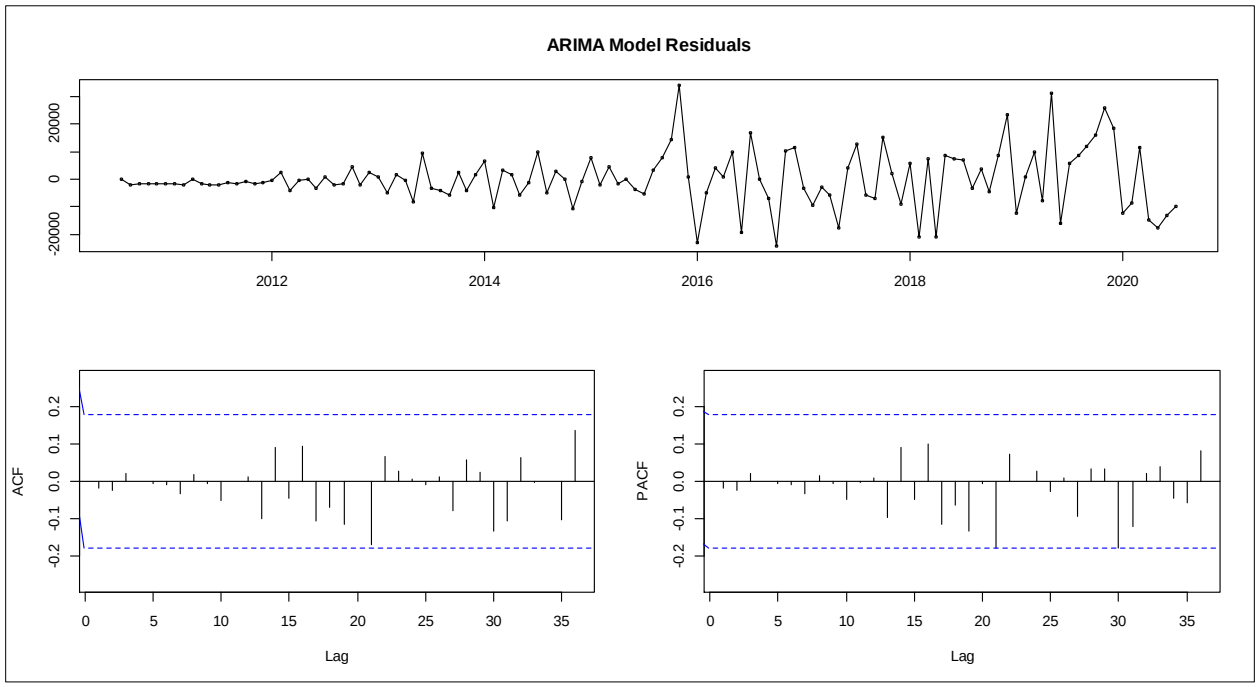


Figure 7-20: Residual of ARIMA model - Kaybob South

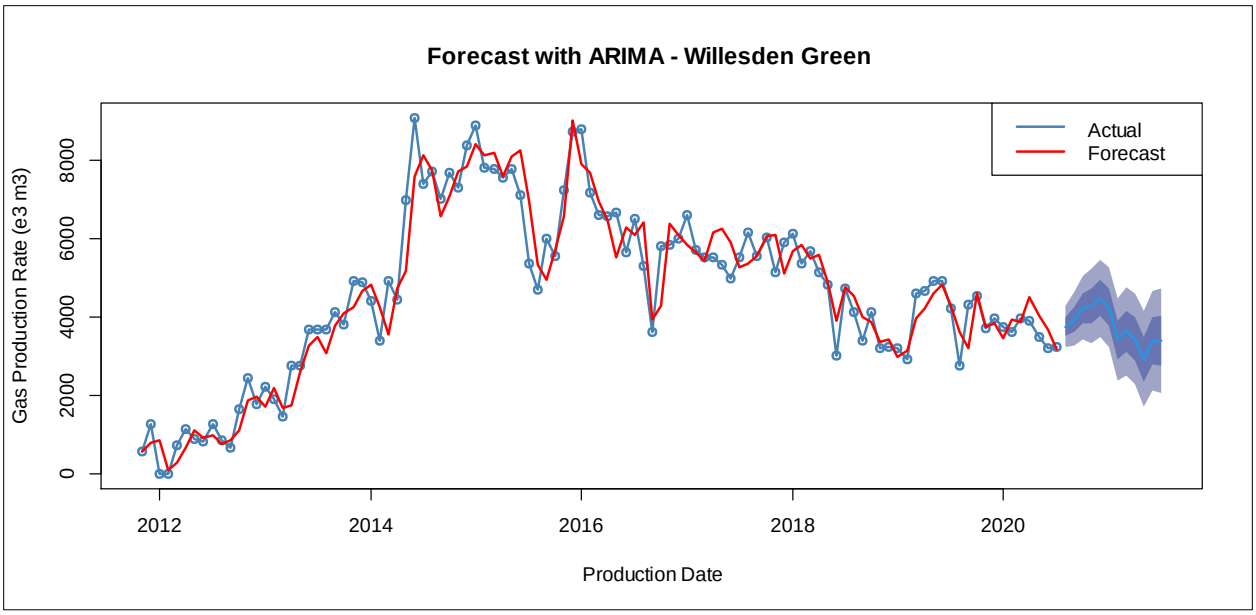


Figure 7-21: Gas production forecasts of Willesden Green field with ARIMA model

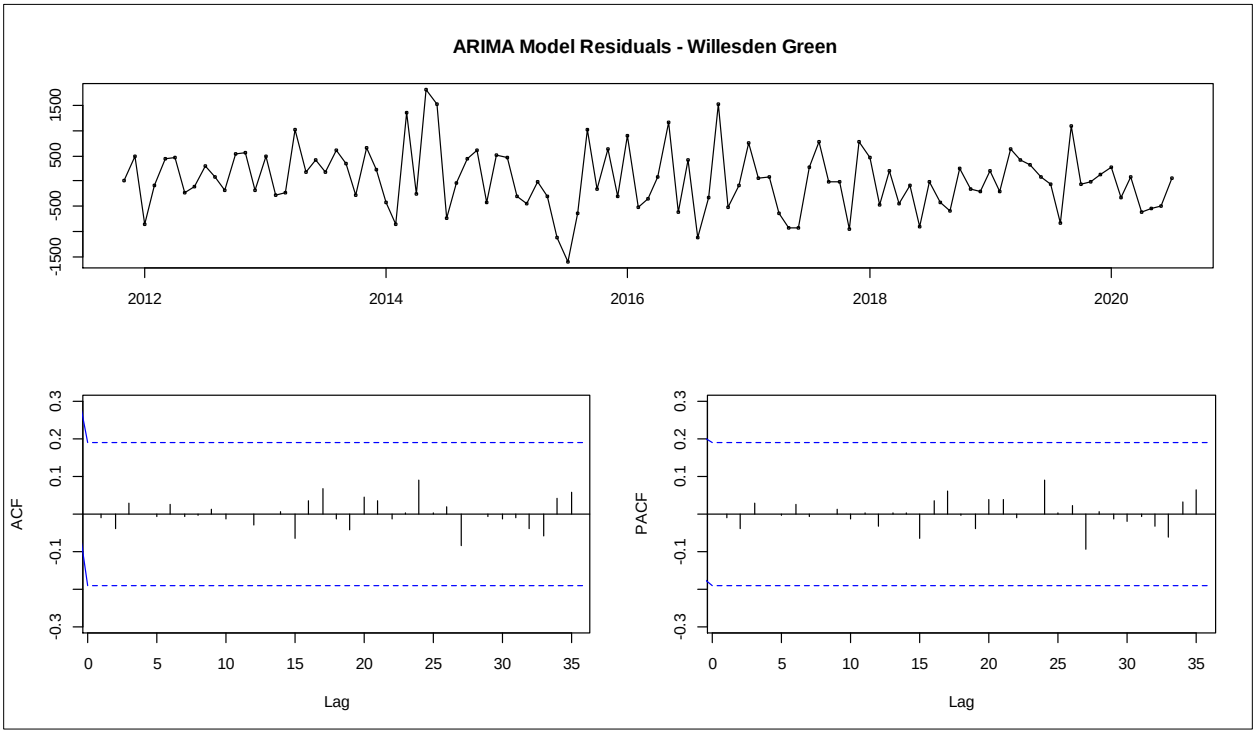


Figure 7-22: Residual of ARIMA model - Willesden Green

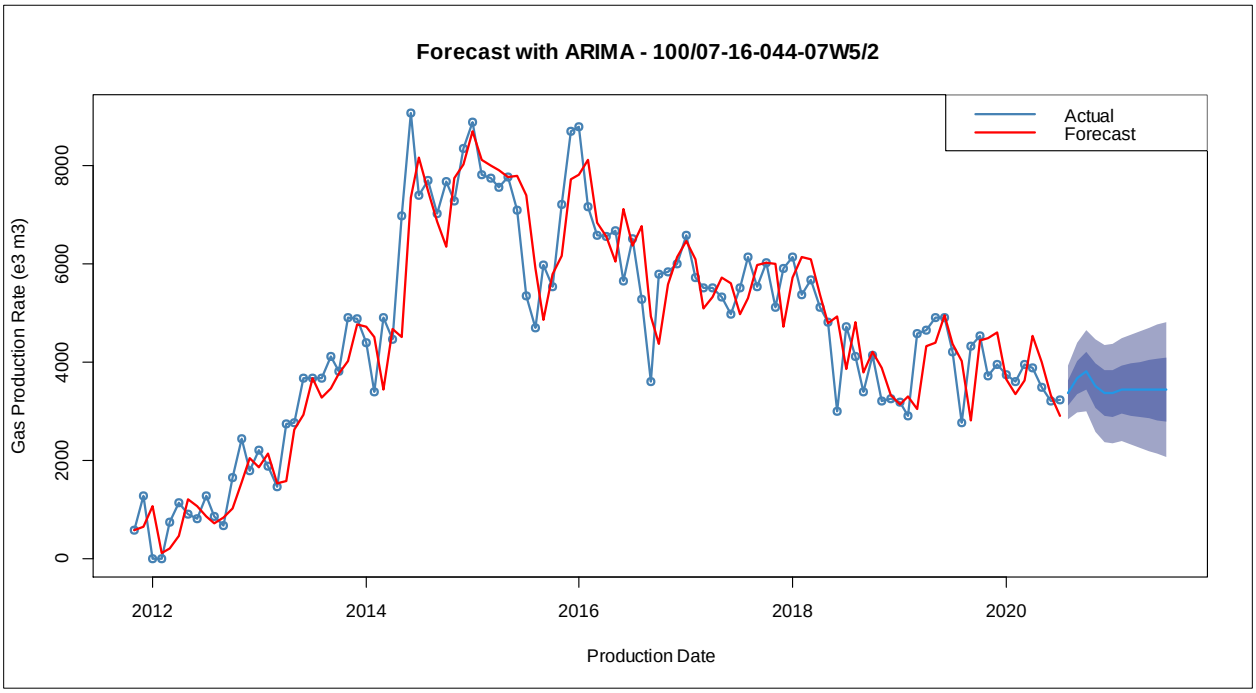


Figure 7-23: Gas production forcast for 100/07-16-044-07W5/2 with ARIMA model

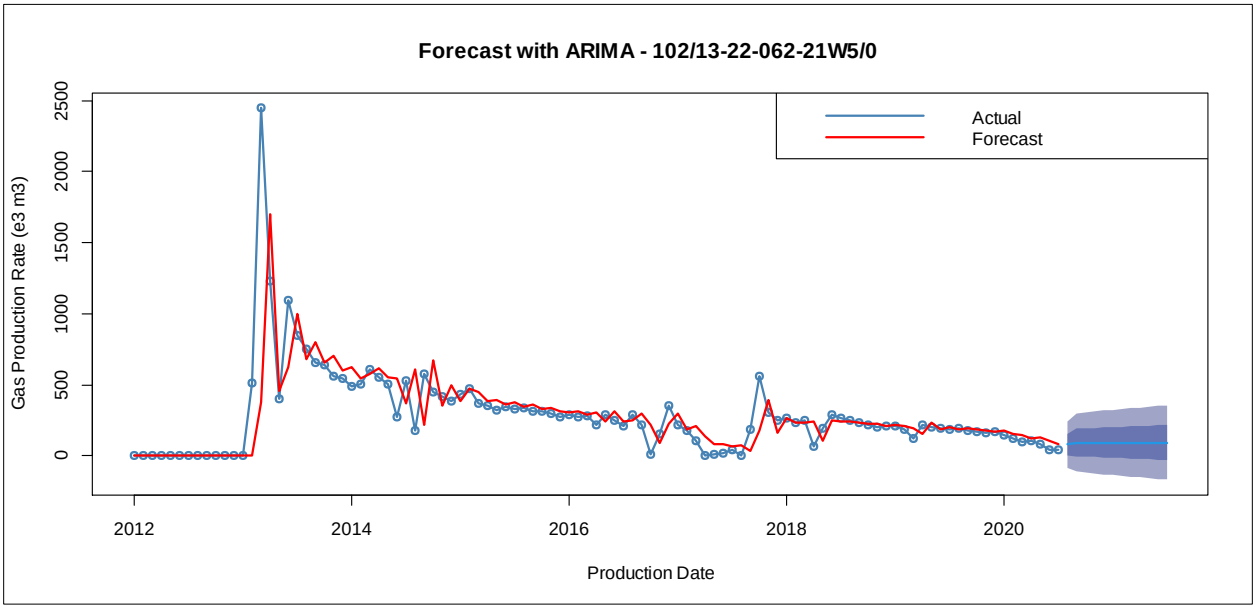


Figure 7-24: Gas production forcast for 102/13-22-062-21W5/0 with ARIMA model

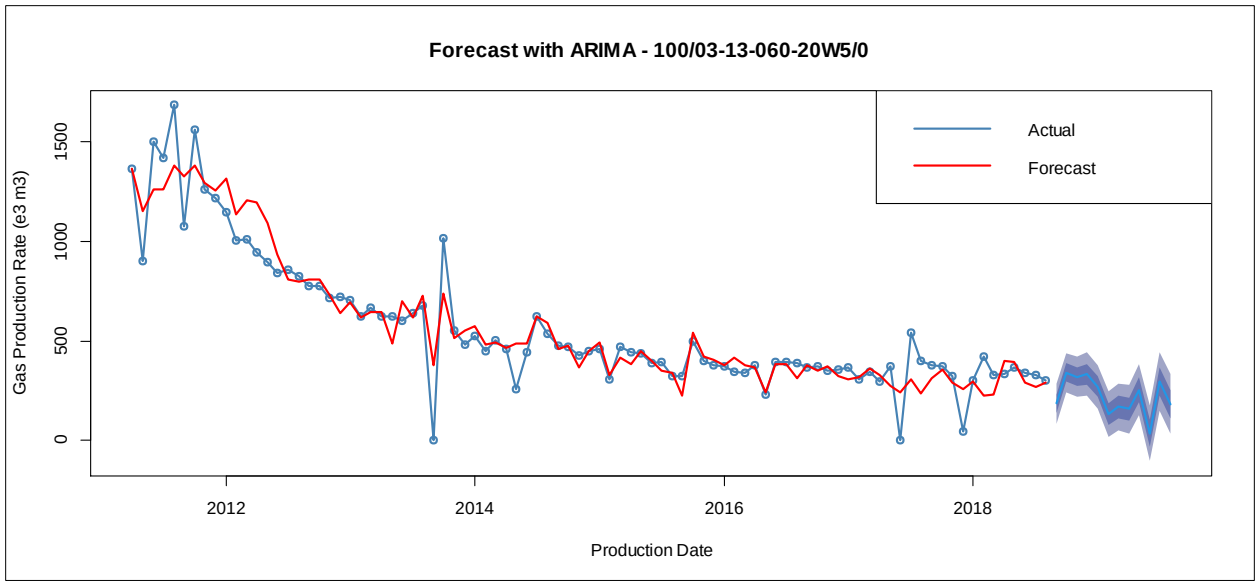


Figure 7-25: Gas production forecast for 100/03-13-060-20W5/0 with ARIMA model

7.3.5 VAR Forecasting Model

One limitation of the ***ARIMA*** model that has been discussed is that they impose a unidirectional relationship, and for gas production, other factors like gas price and total operational time have a critical role. For this purpose, a Vector Autoregression (***VAR***) model would be helpful by considering multiple time series variables.

A ***VAR*** model is a generalization of the univariate autoregressive model for forecasting a vector of time series and extends the idea of univariate autoregression to $\boldsymbol{k}$ time series regression, where the lagged values of all $\boldsymbol{k}$ series appear as regressors. As for ***AR(p)*** models, the lag order is denoted by $\boldsymbol{p}$, so the ***VAR(p)*** model of two variables $\boldsymbol{X_t}$ and $\boldsymbol{Y_t}(\boldsymbol{k = 2})$ is given by the equations

$$Y_t = \beta_{10} + \beta_{11}Y_{t-1} + \cdots + \beta_{1p}Y_{t-p} + \gamma_{11}X_{t-1} + \cdots + \gamma_{1p}X_{t-p} + u_{1t} \qquad (7\text{-}7)$$

$$X_t = \beta_{20} + \beta_{21}Y_{t-1} + \cdots + \beta_{2p}Y_{t-p} + \gamma_{21}X_{t-1} + \cdots + \gamma_{2p}X_{t-p} + u_{2t} \qquad (7\text{-}8)$$

The $\boldsymbol{\beta}$s and $\boldsymbol{\gamma}$s can be estimated using OLS (Ordinary Least Square) on each equation [54]. To build the ***VAR*** model, time-based variables, including gas price, total monthly production hours, and monthly gas production rate from Kaybob-South field converted to time series variables. Figure

7-27 visualize this time series variable. Again, we need to assess if the variables are stationary or not. ***ADF*** test of time series variable shows that all variables are stationary (Figure 7-26). Next, we must select the optimum lag order behind the ***VAR*** like ***ARIMA*(*p*,*d*,*q*)**. The evaluation shows that $\boldsymbol{p = 4}$ is the opimum lag value after checking ***AIC*** and ***BIC*** of different variations. Figure 7-28, Figure 7-29, and Figure 7-30 show the forecasting result for 12-month ahead. The forecasted plots show that gas production is expected to decrease slightly and rebound later on. Gas prices are expected to decrease gradually; however, operational activity is expected to increase.

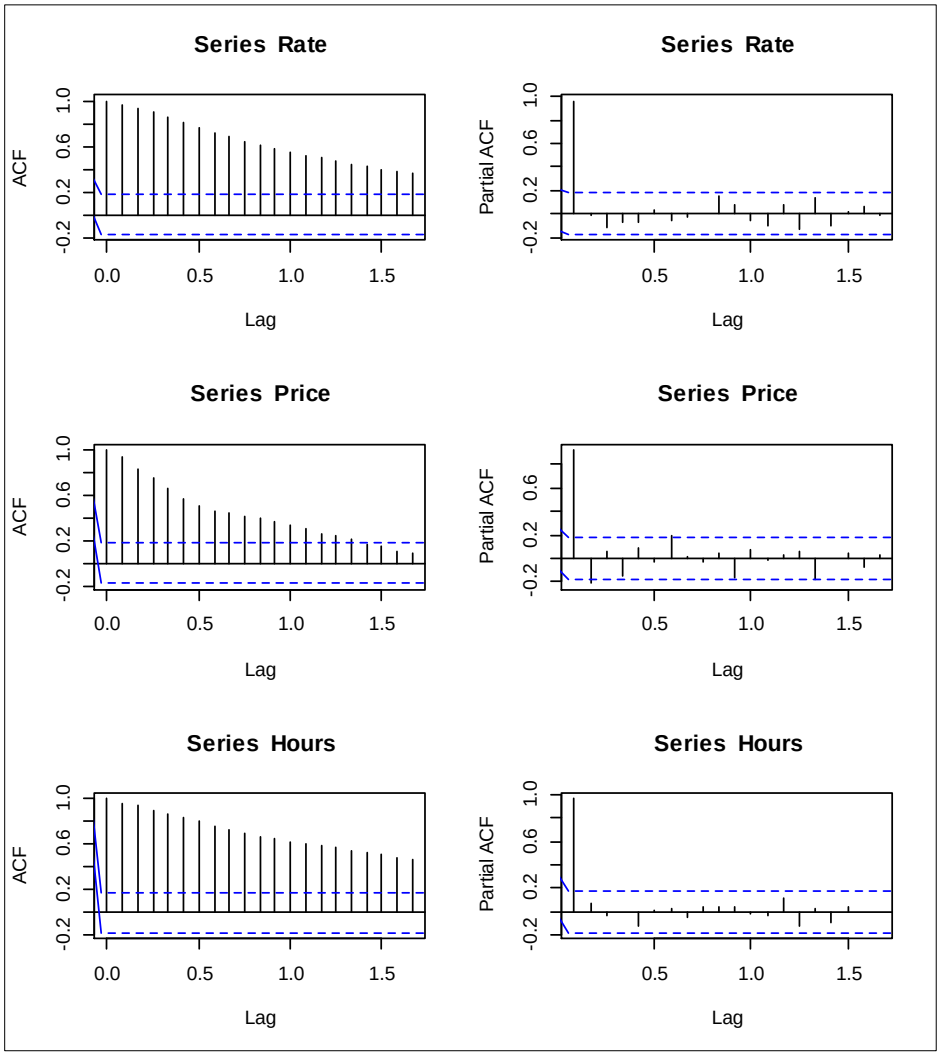


Figure 7-26: Autocorroletion evaluation for time series variables

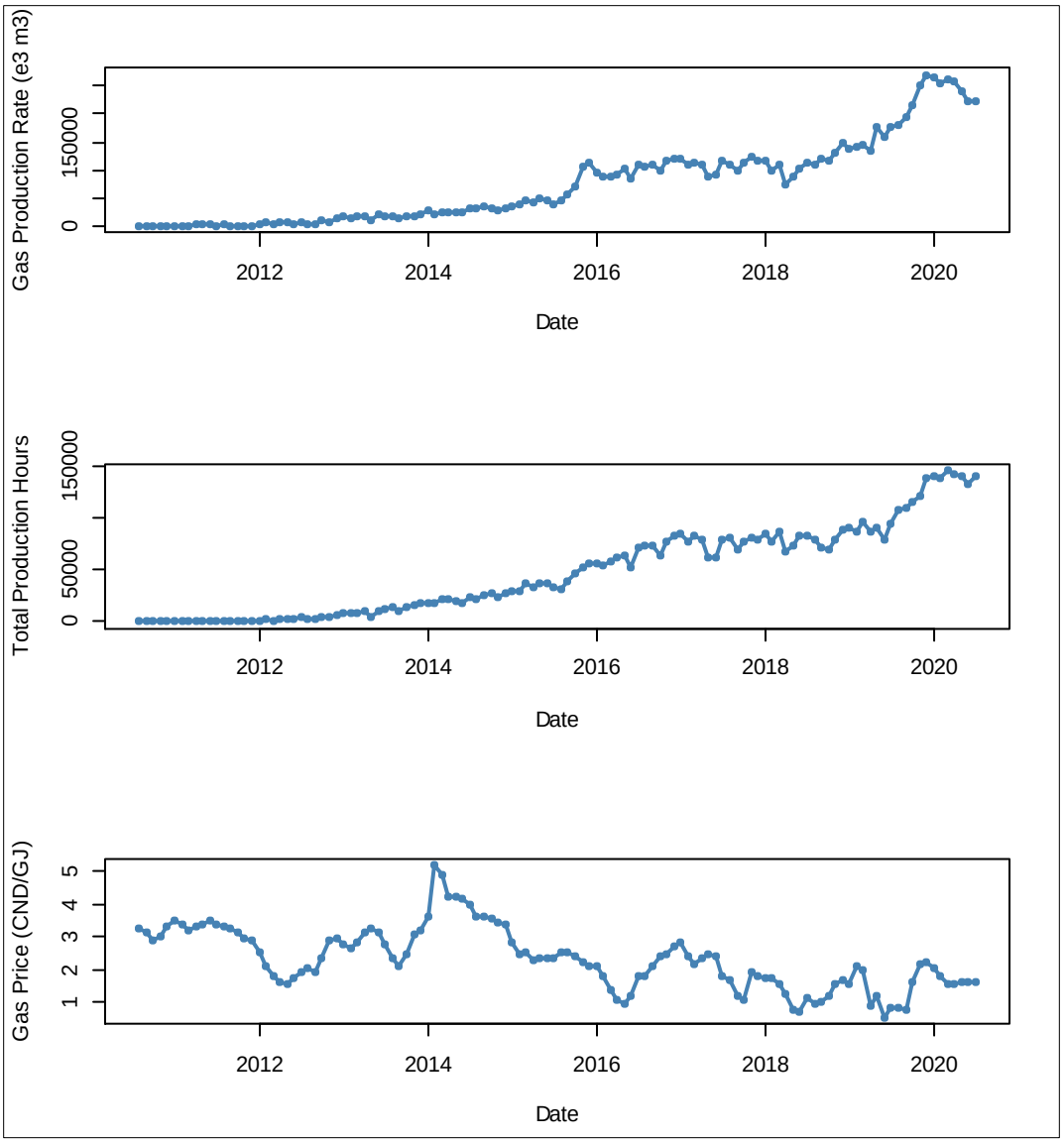


Figure 7-27: Time series variable of VAR model

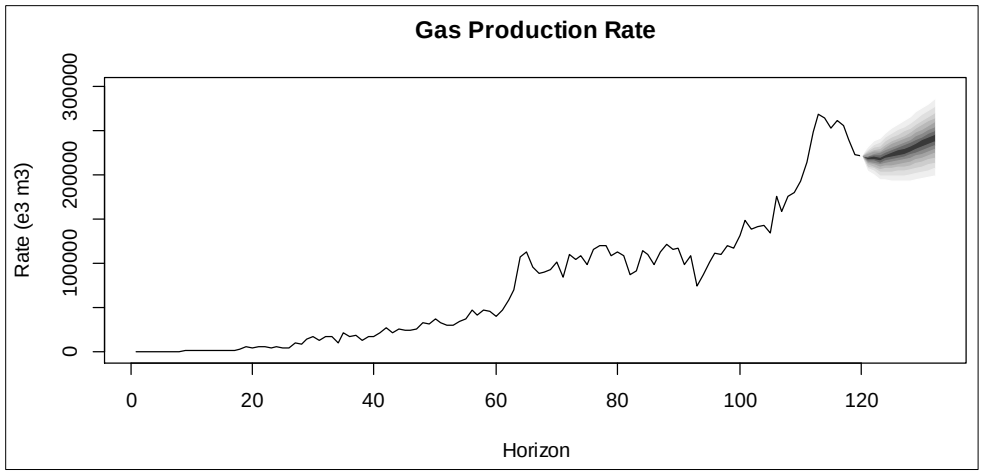


Figure 7-28: VAR prediction for gas production rate for 12-month ahead

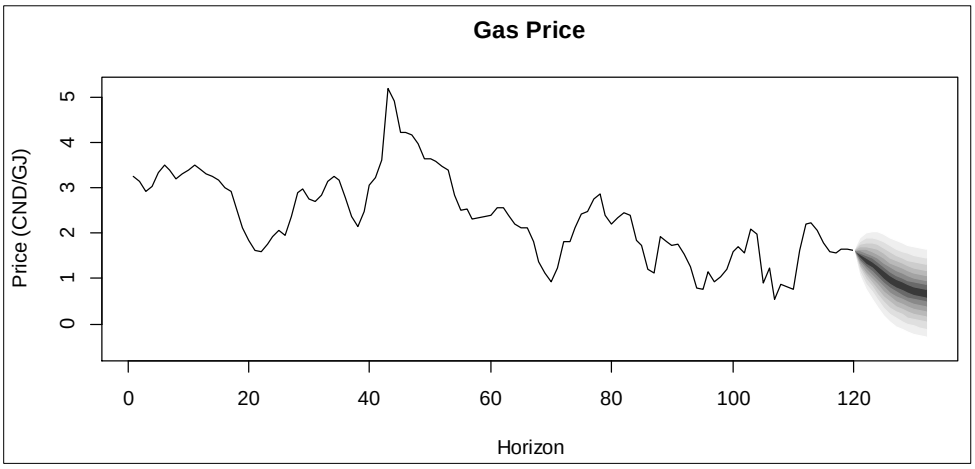


Figure 7-29: VAR prediction for gas price for 12-month ahead

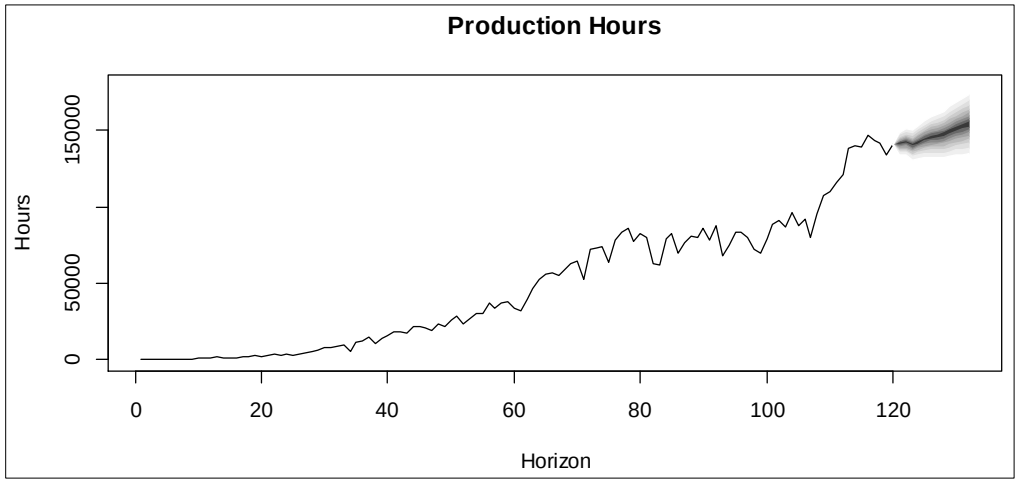


Figure 7-30: VAR prediction for production activities for 12-month ahead

7.4 Machine Learning as Time Series

Time series algorithms are used extensively for analyzing and forecasting time-based data; however, the effect of other factors will be ignored. Here, we want to examine machine learning's ability to add more complexity to the time series by including other factors to understand their impact on forecasting. To include dates into the ML model, we will split dates into different components, including year, month, quarter, and days. A dataset of different variables collected from well, production, drilling and completion, and gas components have been created, and then monthly gas production has been considered a target feature for ML models. After preprocessing, including normalization and imputation, different ML models have been created: Linear Regression, Gradient Boosting, Random Forest, Neural Network, and AdaBoost. To evaluate the performance of different algorithms, 20% of samples were randomly selected for testing, and the remaining sample was used to train the models.

In Figure 7-31, feature importance has used to explain the input features. Figure 7-32 explains which features contribute the most and how they contribute toward the prediction. Features in the plot are ordered by their relevance to the prediction. The production year is the most important feature to estimate monthly gas production.

The result of cross-validation with 10-fold samples has listed in Table 7-2. AdaBoost has the highest ($R^2 = 91\%$) and Linear Regression ($R^2 = 34\%$) the lowest accuracy. To aggregate different models and improve the prediction, all models are stacked with the ensembling method. The result has not shown any considerable improvement in the prediction model.

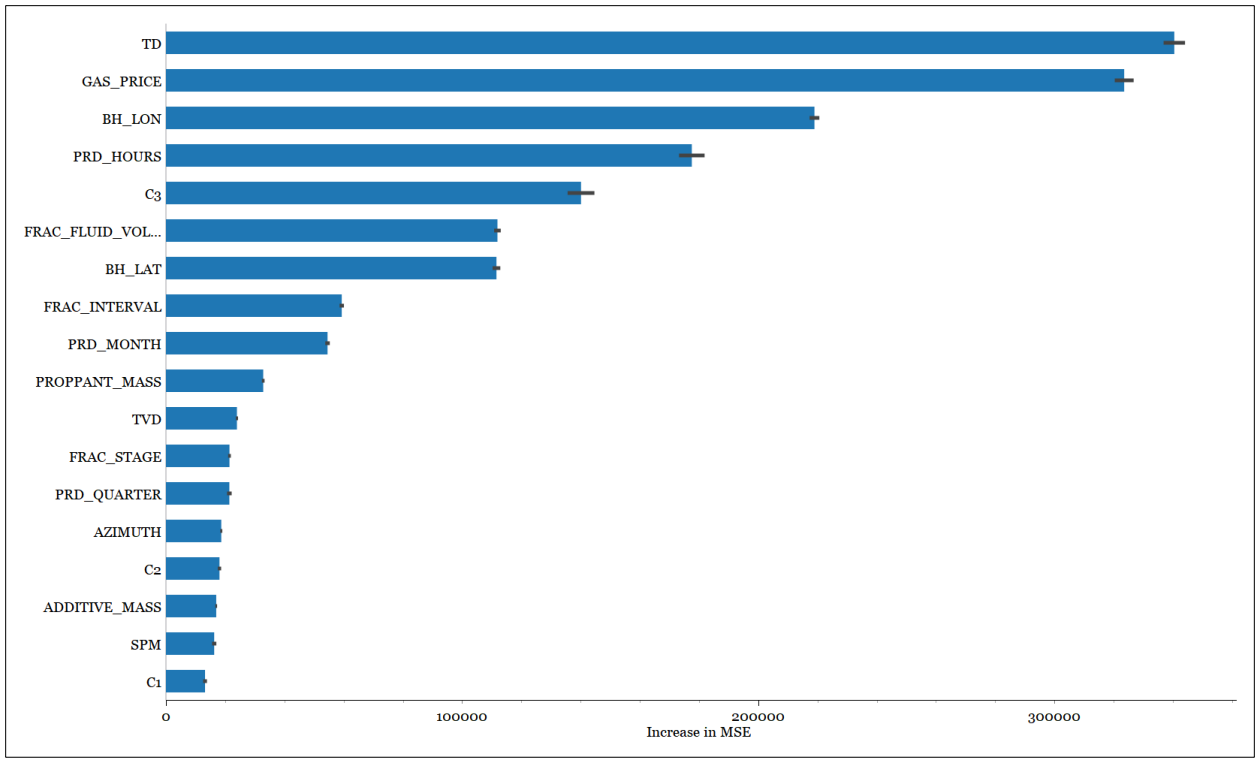


Figure 7-31: Feature importance

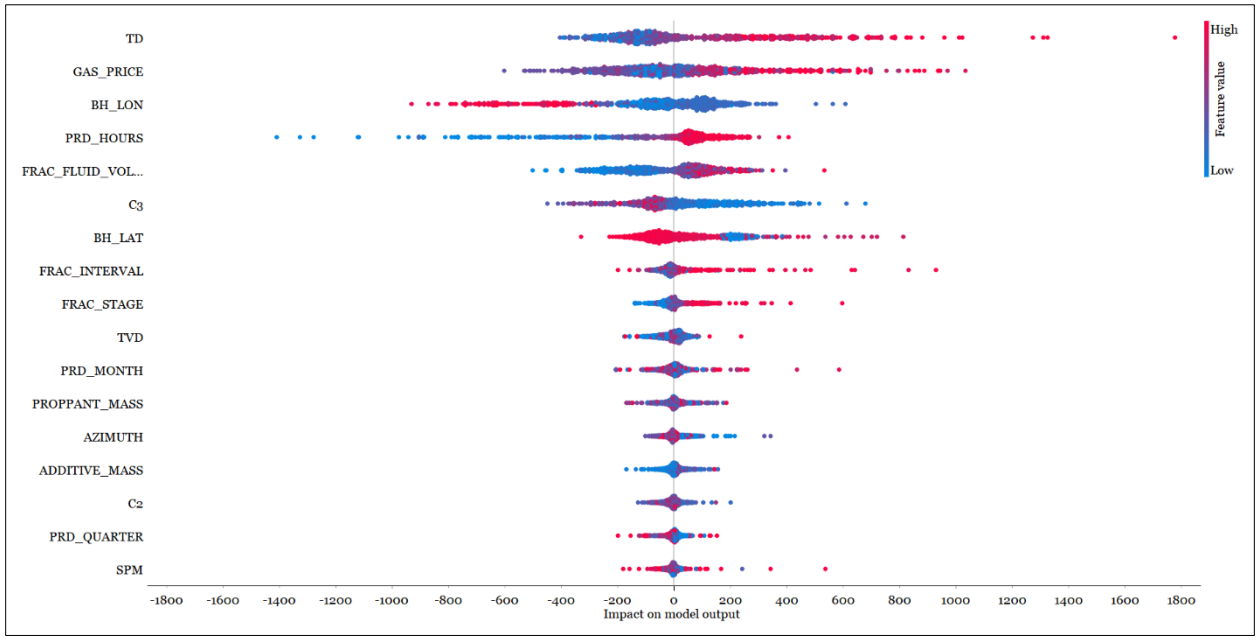


Figure 7-32: Impact of different features on prediction model

Table 7-2: Evaluation result of time-based ML prediction

Model	MSE	RMSE	MAE	R^2
Linear Regression	355187	596	399	0.341
Neural Network	172737	416	260	0.680
Gradient Boosting	170779	413	249	683
Random Forest	56207	237	123	0.896
AdaBoost	48683	221	110	0.910

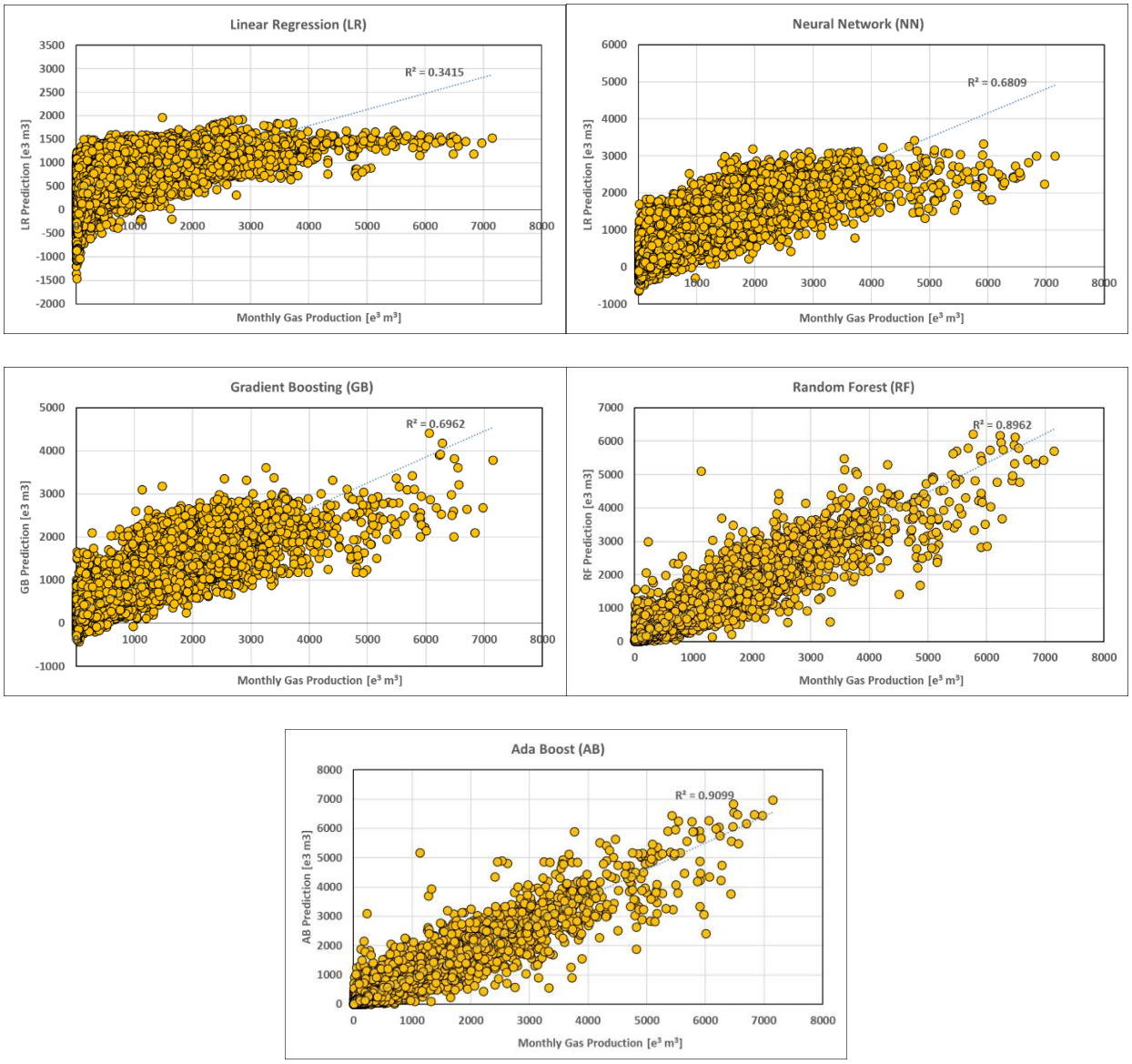


Figure 7-33: correlation plot between predicted data and monthly gas production data

8 CONCLUSION & RECOMMENDATION

In this study, different methods of production forecasting and prediction for shale plays have been reviewed and discussed in detail. These methods include reservoir simulation, traditional decline curve analysis, machine learning, and time series.

Although reservoir simulation is accurate and has a solid physical basis for describing reservoir behaviors and fluid flow mechanisms, it is an expensive and time-consuming process and requires many parameters as input that are not always available. Building and maintaining the model is also cumbersome. Due to the complexity of fluid flow in shale plays and restrictions of reservoir simulators to model the multi-stage hydraulic fractures, it is disputed to accept the final results.

Traditional approaches of decline curves analysis are fast and straightforward with reasonable results in conventional wells; however, they cannot emulate the production mechanisms in shales. If there are enough historical production data, modified and revolved versions of Arps models can acceptedly model the historical data; however, the future production in shale highly depends on the market situation and field operational expenditures (OPEX), which is not considered in this method. DCA depends on the fitting technique and its parameters and is prone to over/underestimation.

Proxy data-driven models have shown their power to mimic reservoir behavior mainly in shale plays, which is still unclear governing physical behavior in wellbores and fields. However, the accuracy and generalization of these black-box models is a place of arguments. Implementing time series-based algorithms provided enough wells and production history data can predict and forecast the future situation of wells and fields.

In this work, first, the impact of different features on the production stream in Duvernay shale assisting by analytical approaches has been studied. Then decline curves are used to estimate future

production rate and EUR in wellbore and field scales. This method failed in the unmatured field but is able to reasonably model the future production in wells with enough production history. However, this approach is highly dependent on fitting methods and their parameters. Overestimation and underestimation are expected problems with this method.

Although machine learning algorithms can successfully describe the hidden non-linearities in data, the risk of inaccurate estimation for unseen samples makes this approach vulnerable. Feeding the model with more samples and implementing more robust algorithms can be a solution to avoid this issue. As shown here in this study, feeding predictive models with time-based features like seasonal features and market situations that directly impact production can make the model smarter in its final decision. Nevertheless, their implementation should be done cautiously and require to be feed with enough field data.

Time series analysis (TSA) and forecasting have revealed promising results in this study and can be an alternative for DCA in shale plays. They are simple but highly accurate and give accurate predictions for future values. They help to identify the patterns and outliers in the data. They can be applied for different scales such as well, field, and operator. As long as there is a trend in historical data, restriction in observations is not a hurdle for forecasting. In contrast, they are helpful for short-term forecasting and could lead to the wrong prediction in long-term forecasting. Multivariate TSA is a more flexible approach than univariate one and can capture the complexity of shale plays.

To improve the prediction performance, it is recommended:

- to update and re-evaluate currently developed models with new field data
- to integrate data-driven models with known physical constraints and guide them to behave logically in unseen situations
- to apply more advanced time series methods like RNN and LSTM for long term forecasting

REFERENCES

1. Serintel. *Reservoir Simulation*. 2021; Available from: http://www.oil-gasportal.com/reservoir-simulation/.

2. Mattax, C.C. and R.L. Dalton, *Reservoir Simulation (includes associated papers 21606 and 21620).* Journal of Petroleum Technology, 1990. **42**(06): p. 692-695.

3. Temizel, C., et al., *A Comprehensive Review of Smart/Intelligent Oilfield Technologies and Applications in the Oil and Gas Industry*, in *SPE Middle East Oil and Gas Show and Conference*. 2019.

4. Alatrach, Y., et al., *Data-Driven Vs. Traditional Reservoir Numerical Models: A Case Study Comparison of Applicability, Practicality and Performance*. 2019.

5. Mohaghegh, S.D., *Shale Analytics : data-driven analytics in unconventional resources*. 2017: Cham, Switzerland : Springer Verlag.

6. Mohaghegh, S.D., *Reservoir simulation and modeling based on artificial intelligence and data mining (AI&DM).* Journal of Natural Gas Science and Engineering, 2011. **3**(6): p. 697-705.

7. Arps, J.J., *Analysis of Decline Curves.* Transactions of the American Institute of Mining and Metallurgical Engineers, 1945. **160**(01): p. 228-247.

8. Belyadi, H., E. Fathi, and F. Belyadi, *Hydraulic fracturing in unconventional reservoirs: theories, operations, and economic analysis*. 2019: Gulf Professional Publishing.

9. Belyadi, H., E. Fathi, and F. Belyadi, *Chapter Seventeen - Decline curve analysis*, in *Hydraulic Fracturing in Unconventional Reservoirs (Second Edition)*, H. Belyadi, E. Fathi, and F. Belyadi, Editors. 2019, Gulf Professional Publishing. p. 311-340.

10. Doublet, L.E., et al., *Decline Curve Analysis Using Type Curves--Analysis of Oil Well Production Data Using Material Balance Time: Application to Field Cases*. 1994.

11. Guo, B., *Petroleum production engineering, a computer-assisted approach*. 2011: Elsevier.

12. Shahamat, M.S.S., L. Mattar, and R. Aguilera, *Analysis of Decline Curves on the Basis of Beta-Derivative.* SPE Reservoir Evaluation & Engineering, 2015. **18**(02): p. 214-227.

13. Mattar, L., *Production Analysis and Forecasting of Shale Gas Reservoirs: Case History-Based Approach*. 2008.

14. Lee, J. and R. Sidle, *Gas-Reserves Estimation in Resource Plays.* SPE Economics & Management, 2010. **2**(02): p. 86-91.

15. Robertson, S., *Generalized hyperbolic equation.* 1988.

16. Seshadri, J. and L. Mattar, *Comparison of Power Law and Modified Hyperbolic Decline Methods*. 2010.

17. Ilk, D., et al., *Exponential vs. Hyperbolic Decline in Tight Gas Sands: Understanding the Origin and Implications for Reserve Estimates Using Arps' Decline Curves*. 2008.

18. Valkó, P.P. and W.J. Lee, *A Better Way to Forecast Production from Unconventional Gas Wells*. 2010.

19. Duong, A.N., *An Unconventional Rate Decline Approach for Tight and Fracture-Dominated Gas Wells*. 2010.

20. Clark, A.J., *Decline curve analysis in unconventional resource plays using logistic growth models.* 2011.

21. Zhang, H., et al., *Extended exponential decline curve analysis.* Journal of Natural Gas Science and Engineering, 2016. **36**: p. 402-413.

22. Mishra, S., *Exploring the Diagnostic Capability of RTA Type Curves*. 2014.

23. Tadjer, A., A. Hong, and R.B. Bratvold, *Machine learning based decline curve analysis for short-term oil production forecast.* Energy Exploration & Exploitation, 2021.

24. Johnson, D. *Machine Learning Tutorial for Beginners: What is, Basics of ML*. 2021; Available from: https://www.guru99.com/machine-learning-tutorial.html.

25. Brownlee, J. *Difference Between Algorithm and Model in Machine Learning*. 2020; Available from: https://machinelearningmastery.com/difference-between-algorithm-and-model-in-machine-learning/.

26. Heller, M. *Machine learning algorithms explained*. 2019; Available from: https://www.infoworld.com/article/3394399/machine-learning-algorithms-explained.html.

27. Bhattacharjee, J. *Some Key Machine Learning Definitions*. 2017; Available from: https://medium.com/technology-nineleaps/some-key-machine-learning-definitions-b524eb6cb48.

28. Brownlee, J. *What is the Difference Between a Parameter and a Hyperparameter?* 2019; Available from: https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/.

29. Brownlee, J. *Difference Between Classification and Regression in Machine Learning*. 2017; Available from: https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/.

30. Brownlee, J. *10 Clustering Algorithms With Python*. 2020; Available from: https://machinelearningmastery.com/clustering-algorithms-with-python/.

31. Taranenko, L. *How to Apply Machine Learning to Demand Forecast*. 2021; Available from: https://mobidev.biz/blog/machine-learning-methods-demand-forecasting-retail.

32. Ng, A. *Introduction to Regression Analysis, and Gradient Descent*. 2021; Available from: http://www.holehouse.org/mlclass/index.html.

33. Kelleher, J.D., *Deep learning*. 2019: MIT press.

34. Sharma, S. *Activation Functions in Neural Networks*. 2017; Available from: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

35. AI-Wiki. *Activation Function*. 2021; Available from: https://docs.paperspace.com/machine-learning/wiki/activation-function.

36. Cortes, C. and V. Vapnik, *Support-vector networks.* Machine learning, 1995. **20**(3): p. 273-297.

37. *Gradient Boosting in Machine Learning*. 2020; Available from: https://intellipaat.com/blog/gradient-boosting-in-machine-learning/.

38. Tin Kam, H. *Random decision forests*. in *Proceedings of 3rd International Conference on Document Analysis and Recognition*. 1995.

39. Breiman, L., *Random Forests.* Machine Learning, 2001. **45**(1): p. 5-32.

40. Sidhu, R. *Use of Cross-Validation in Machine Learning*. 2019; Available from: https://aigraduate.com/use-of-cross-validation-in-machine-learning/.

41. Brownlee, J., *Ensemble Learning Algorithms With Python: Make Better Predictions with Bagging, Boosting, and Stacking*. 2021: Machine Learning Mastery.

42. Peiris, M., et al., *Detecting Behavioral Microsleeps from EEG Power Spectra.* Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference, 2006. **1**: p. 5723-6.

43. Palit, A.K. and D. Popovic, *Computational intelligence in time series forecasting: theory and engineering applications*. 2006: Springer Science & Business Media.

44. Weigend, A.S., *Time series prediction: forecasting the future and understanding the past*. 2018: Routledge.

45. De Gooijer, J.G. and R.J. Hyndman, *25 years of time series forecasting.* International journal of forecasting, 2006. **22**(3): p. 443-473.

46. Hamzaçebi, C., D. Akay, and F. Kutay, *Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting.* Expert systems with applications, 2009. **36**(2): p. 3839-3844.

47. Sorjamaa, A. and A. Lendasse. *Time series prediction using DirRec strategy*. 2006. Citeseer.

48. Kline, D.M., *Methods for multi-step time series forecasting neural networks*, in *Neural networks in business forecasting*. 2004, IGI Global. p. 226-250.

49. Taieb, S.B., et al. *Long-term prediction of time series by combining direct and mimo strategies*. 2009. IEEE.

50. Taieb, S.B., et al., *A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition.* Expert systems with applications, 2012. **39**(8): p. 7067-7083.

51. Bontempi, G., S.B. Taieb, and Y.-A. Le Borgne. *Machine learning strategies for time series forecasting*. 2012. Springer.

52. Redell, N., *forecastML: Time Series Forecasting with Machine Learning Methods. R package version 0.9. 0*. 2020.

53. Svensén, M. and C.M. Bishop, *Pattern recognition and machine learning*. 2007, Springer.

54. Hyndman, R.J. and G. Athanasopoulos, *Forecasting: principles and practice*. 2018: OTexts.

55. Preston, A., et al., *Duvernay Reserves and Resources Report: A Comprehensive Analysis of Alberta's Foremost Liquids-Rich Shale Resource*. 2016: Alberta Energy Regulator.

56. Rokosh, C.D., et al., *Summary of Alberta's shale-and siltstone-hosted hydrocarbon resource potential.* Energy Resources Conservation Board, ERCB/AGS Open File Report, 2012. **6**: p. 327.

57. Magoon, L.B., *The petroleum system—a classification scheme for research, exploration, and resource assessment.* Petroleum systems of the United States: US Geological Survey Bulletin, 1988. **1870**: p. 2-15.

58. Mossop, G.D. and I. Shetsen, comp. *Geological atlas of the Western Canada Sedimentary Basin*. 1994; Available from: https://ags.aer.ca/reports/atlas-western-canada-sedimentary-basin.

59. *Duvernay Petroleum System (GIS data, line features)*. 2016: Alberta Energy Regulator.

60. Accumap, *Wells Data Completed in Duvernay Formation*. 2020: IHS Markit.

61. Heidbach, O., et al., *The World Stress Map database release 2016: Crustal stress pattern across scales.* Tectonophysics, 2018. **744**: p. 484-498.

62. *Oil and Natural Gas Price*. 2021: Government of Alberta.

63. Fuller, W.A., *Introduction to statistical time series*. Vol. 428. 2009: John Wiley & Sons.

64. Srivastava, T. *A Complete Tutorial on Time Series Modeling in R*. 2015; Available from: https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/.