بسم الله الرحمن الرحيم

Sudan university of science and technology

College of petroleum & mining engineering

Department of petroleum engineering

Machine Learning Approach for Water Control Diagnostics Plots

نظام التعلم الآلي للمخططات التشخيصية للتحكم في إنتاج المياه

A Research Submitted to College of Petroleum & Mining Engineering in Partial Fulfillment of the Requirements for the Degree of B.Sc. in Petroleum Engineering

Prepared by:

Abdulhaleem Mohammed Abbas Ahmed.

Ahmed alkhatim Awad eljeed.

Ammar Mohmmed Abdullah Musa.

Mohammed Hamid Ahmed.

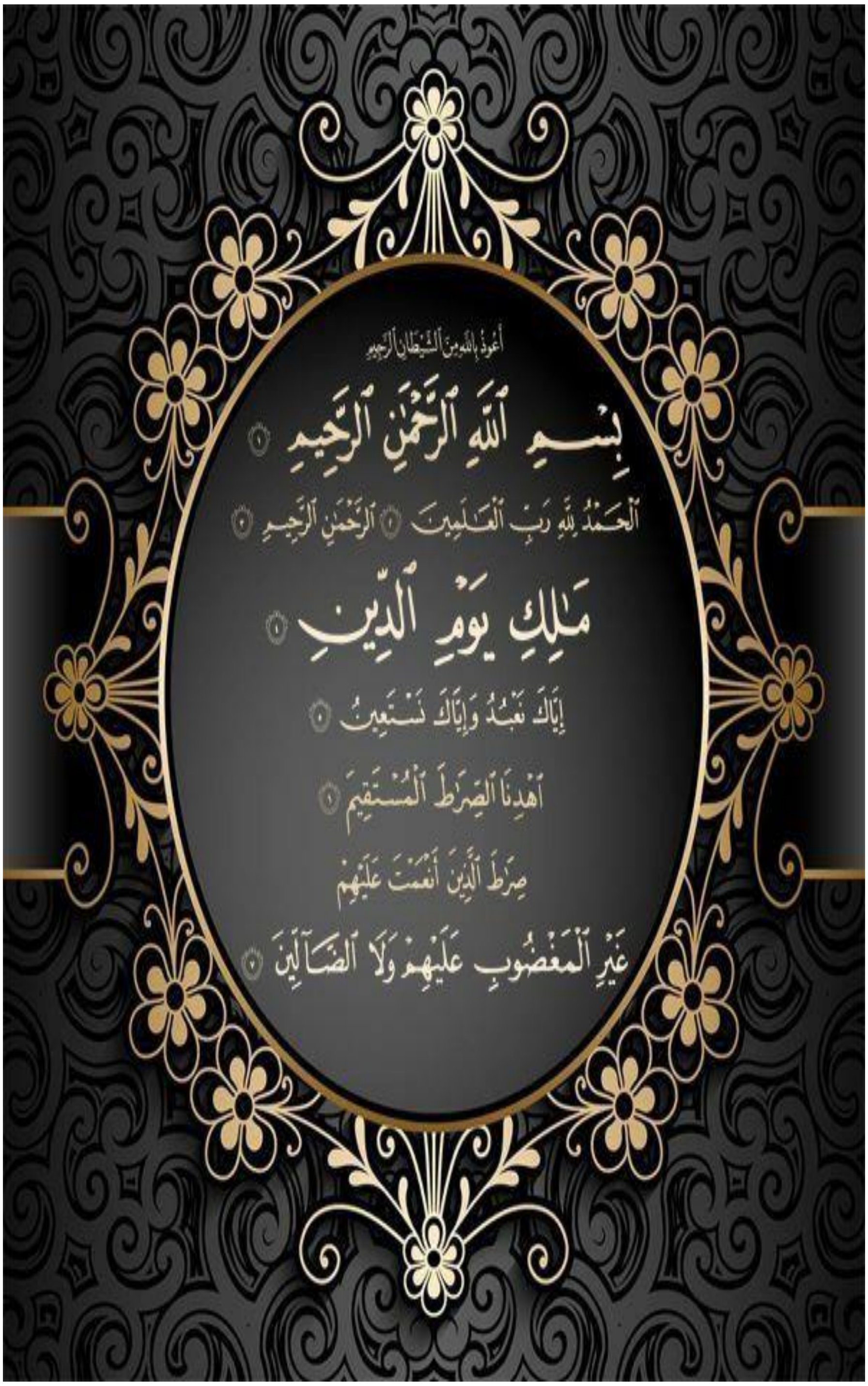
Samreen Mohammed najib Yassin.

Supervised by:

T. Muhanned Mahgoup Mohammed khairy.

February 2022

الإستهلال


أعوذ بالله من الشيطان الرجيم
بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ
ٱلْحَمْدُ لِلَّهِ رَبِّ ٱلْعَٰلَمِينَ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ
مَٰلِكِ يَوْمِ ٱلدِّينِ
إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ
ٱهْدِنَا ٱلصِّرَٰطَ ٱلْمُسْتَقِيمَ
صِرَٰطَ ٱلَّذِينَ أَنْعَمْتَ عَلَيْهِمْ
غَيْرِ ٱلْمَغْضُوبِ عَلَيْهِمْ وَلَا ٱلضَّآلِّينَ

<u>Dedication</u>

This unpretentious effort is wholeheartedly dedicated to our beloved **parents** who have been a source of inspiration and motive for every good step in our life, who continuously provide their spiritual, emotional and financial support.

To our **brothers, sisters, relative, mentor, friends** and **classmates** who their motivation and encouragement to finish this work.

And lastly for **people who know the greatness of knowledge,** but do not have ability to obtain it this work is dedicated to you.

Researchers…

Acknowledgment

We would like to acknowledge and give our warmest thanks to our supervisor **T. Muhanned Mahgoup khairy** who made this work possible. His guidance and advice carried us through all the stages of writing this project. We would also like to thank our committee members for letting our defense be an enjoyable moment, and for their brilliant comments and suggestion.

THANK YOU

Abstract

Chan's water control diagnostic plots are the most common way to investigate the mechanism that causes water production, but the traditional investigation by human has a degree of uncertainty and requires long time. The purpose of this project is to build a supervised machine learning model using ridge classifier to detect water production mechanism WPM (coning or channeling) accurately and time effectively.

Firstly, we performed a conventional identification of Chan's plots pattern from Heglig's production data, then the data is divided into training set and test set, also the training set is split into two trends to train two different Models and create the ensemble classifier, then we evaluate the model ability on the test set, and then a hyper parameter (alpha) was tested many times to improve each model accuracy.

We find out that Model1 demonstrates high accuracy to detect the WPM, and it is able to overfit the training set by 100% also it showed high degree of accuracy to classify unseen data (generalization) by 100% and for Model2 the overfitting is 100% and generalization is 89%, The accuracy of ensemble classifier is 100% on the test set.

The project showed successful application of Machine Learning by using ensemble classifier and Ridge Classification algorithm to classify the WPM based on Chan's water control diagnostic plots in efficient way that would make the WPM detection much easier. Some other mechanisms may be included in the future work that would be done to develop this model.

التجريد

مخططات Chan هي مخططات شائعة جداً لتشخيص آلية إنتاج المياه ولكن التشخيص التقليدي عن طريق العين المجردة يحدث فيه بعض اللبس وأحيانا يصعب إتخاذ القرار إضافة إلى الوقت الكثير الذي يتطلبه. يهدف هذا المشروع إلى بناء نموذج (تعليم الآلة) بإستخدام خوارزمية ال Ridge Classifier للتنبؤ بآلية إنتاج المياه بدقه عالية وزمن قياسي.

.في البداية قمنا بدراسة أنماط Chan لبيانات إنتاج من حقل هجليج ثم قسمنا البيانات إلى قسمين، أحدهم للأختبار والآخر للتدريب الذي بدوره قسم إلي قسمين من المنحنيات المتشابهة الى حد كبير لتدريب نموذجين مختلفين لتكوين (المصنف الجامع) و آخيراً تم دراسة أثر تقليل قيمة hyper parameter على دقة كل نموذج.

أظهر النموذج الأول دقة عالية في تشخيص آلية إنتاج المياه حيث وصل ال overfitting لنسبة 100% وال generalization لنسبة 100% أما النموذج الثاني فقد وصل فيه ال overfitting لنسبة 100% وال generalization لنسبة 89% ،أما الدقة النهائية (للمصنف الجامع) فقد وصلت ألى 100%.

المشروع أظهر تطبيقاً ناجحا لاستخدام مفهوم تعليم الآلة وخوارزمية ال Ride Classifier في تصنيف آلية انتاج المياه بناءً على مخططات Chan والذي ستجعل التصنيف سهل وسريع جدا. بعض الآليات الأخرى يمكن أن تتضمن في العمل المستقبلي لتطوير هذا النموذج.

CONTENTS

List of Figures:

List of Tables:

NOMENCLATURE

WOR = Water Oil Ratio

WOR’ = Water Oil Ratio derivative

WPM = Water Production Mechanism

GOR = Gas Oil Ratio

OWC = Oil Water Contact

WC = Water Cut

ML = Machine Learning

CSV = Comma Separated Values

LMT = Logistic Model Tree

GUIs = Graphical User Interface

SQL = Structure Query Language

SVM = Support Vector Machine

PCA = Principal Component Analysis

NAN = Not a Number

Chapter (1)

Introduction

Excessive water production is one of the main well-known problems that would face any oil operator in the world. Although this problem is typical in older wells, it can also occur in new developed wells as well (Taha & Amani, 2019). Water production is inevitable in most oil wells, Water may also be produced into the wellbore commingled with oil, at a rate below the economic water-oil ratio (WOR) limit, which may not be reduced or shut off without affecting the oil production. This is the so-called "sweep" and "good" water. However, a water production rate exceeding the economic WOR limit or a sudden outburst of water in to the oil well indicates a problem and requires immediate attention so called "unwanted" and "bad" water, it causes numerous economic problems for oil production companies (Rabiei, et al., 2009). First, excessive water effects the performance of the production wells and shortens their lifespan. The presence of the water in the wellbore increases the weight of the fluid column which leads to an increase in the lifting requirements. That increases the operating cost and leads to a lower the drawdown, water production also enhances the presence of scales, corrosion, and degradation in the field facilities starting from the wellbore to the surface facilities. Another major problem is that the cost of separating, treating, and disposing the produced water is a great burden to oil company budgets (Taha & Amani, 2019).

1.1 Sources of Unwanted Water Production:

1.1.1Water flooding:

the fluid tends to take the paths least resistance and the injected water, as a result, goes to the open fractures and high permeability formations instead of matrix rock to displace the oil. In some cases, the water injection well happens to be connected with the production well through an open fracture or features which are known also as 'thief zones (Figure 1:1).

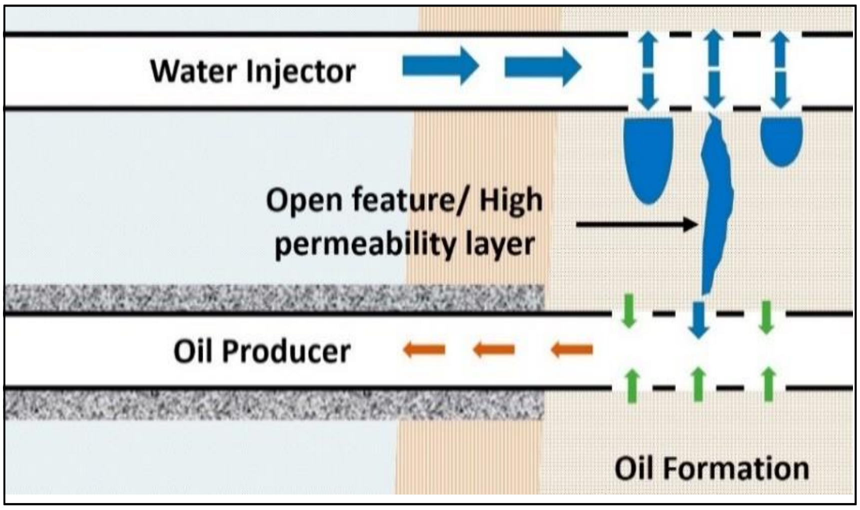


Figure 1.1: Example of a water injection well connected to an oil producer well (Taha & Amani, 2019)

1.1.2 Aquifer:

Open features also can result in an excessive amount of water if they are connected to the aquifer (Figure 1:2).

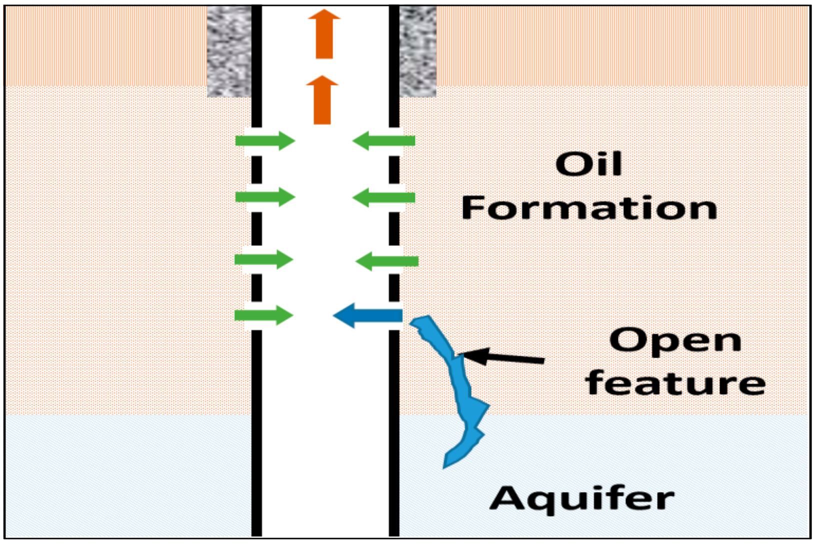


Figure 1.2: An oil producer connected to an aquifer through an open feature (Taha & Amani, 2019)

1.1.3 Formation (connate) water:

Fractures and open features can contribute to unwanted water production

when they are connected to water formations zones (Taha & Amani, 2019).

1.1.4 Gas hydrate reservoirs

Can be also a main source of excessive water production when dissociated**.**

1.2 Mechanisms of Water Production:

1.2.1 Coning:

The term coning is used because, in a vertical well, the shape of the interface when a well is producing the second fluid resembles an upright or inverted cone. Water coning is a term used to describe the mechanism underlying the upward movement of water into the perforations of a producing well (Moradi, et al., 2010).

It is a production-related-problem in partially perforated wells, that is, wells completed at the upper parts of the reservoir. During production of oil, the pressure drops in the well tends to draw-up water from the aquifer towards the lowest completion interval at the well; as shown in Figure (1:3) This rising up of aquifer content - water, is caused by potential distribution near the wellbore.

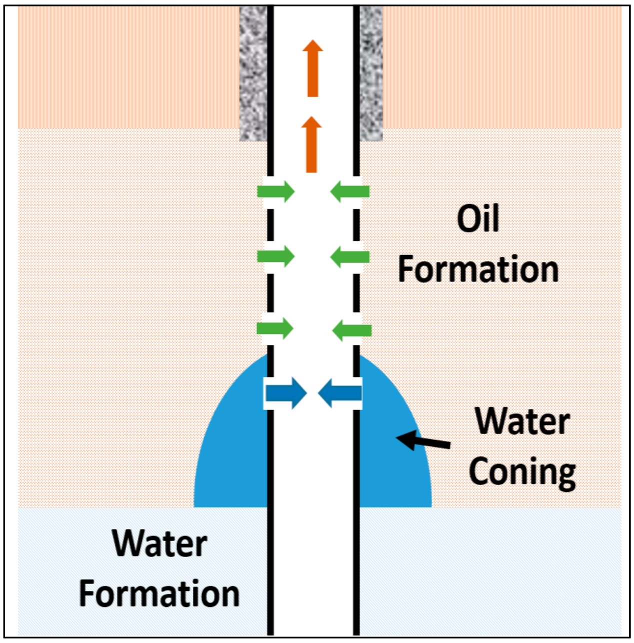
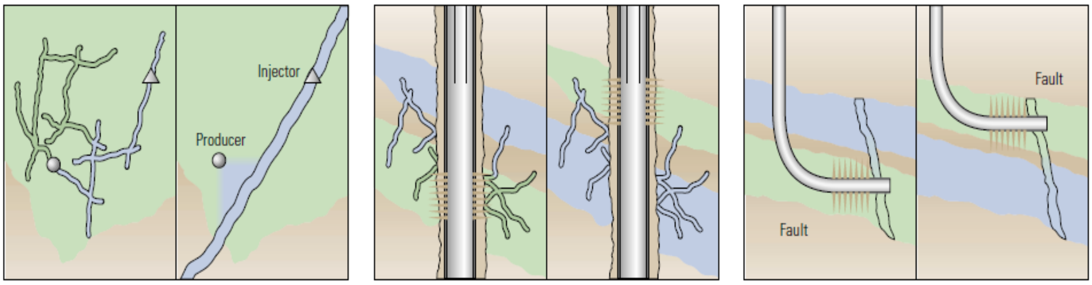


Figure 1.3: water coning (Okon, et al., 2017)

1.2.2. Channeling:

Channeling occurs because of the early breakthrough in the high permeability or fractured formations especially in water flooding. Channeling is one of the more important excessive water productions. Furthermore, reservoir heterogeneities lead to the presence of high permeability streaks. Fractures or fracture-like features are the most common cause of the channeling. Water production could emanate via natural fractures from underlying aquifers. Induced or natural fracture fractures can cause channeling between wells in un fractured reservoir often stratification and associated permeability variations among various layers can result in channeling between an injector and producer or from an edge water aquifer to the producers (Excessive Water Production Diagnosis and Strategies Analysis -Case Study- Jake Field -Sudan, 2016).



^ Fractures or faults between an injector and a producer.

^ Fractures or faults from a water layer (vertical well).

^ Fractures or faults from a water layer (horizontal well).

Figure 1.4: water channeling. (Bailey, et al., 2000)

1.3 Problem Statement:

An important problem in water control is the identification of the dominant reservoir or production mechanisms, The water production mechanism must be properly investigated and accurately diagnosed in order to design an appropriate and effective treatment method. Incorrect, inadequate, or lack of proper diagnosis usually leads to ineffective water control treatments that cost a lot of time and money.

However, some investigations are proper and complete but they are usually made at late times after the breakthrough occurs which leads to economic and operational costs, therefor early investigations should be performed to avoid the problems of late investigation.

1.4 Research Objectives:

This research aimed to:

1. Analyze WOR and WOR’ curves using Chan’s method.

2. perform conventional investigation for some wells production data from Heglig Oil field.

3. Build a machine learning model to Perform early Prediction for excessive water production mechanisms.

Chapter (2)

Literature review

2.1 Water Production Diagnostic:

In the past, water control was simply a plug and cement operation, or a gel treatment in a well. The main reason for why industry's failure to consistently control water is the lack of understanding the different problems and the consequent applications of inappropriate solutions. This is demonstrated by the number of technical papers discussing the treatments and results with little or no reference to the geology, reservoir or water control problem. The key optimum way for water control, is the diagnostics to identify the specific water problem at hand. Well diagnostics are used in three ways:

I. Screening wells that are suitable candidates for water control.

II. Determine the water problem so that a suitable water-control method can be selected.

III. Locate the water entry point in the well so that a treatment can be correctly placed (Bailey, et al., 2000).

2.1.1 Diagnostics with the Production Data:

2.1.1.1 Recovery plot:

The recovery plot is a semi log plot of WOR against cumulative oil production. the production trend can be extrapolated to the WOR economic limit to determine the oil production that will be achieved if no water-control action is taken. If the extrapolated production is approximately equal to the expected reserves for a well, then the well is producing acceptable water, and no water control is needed. If this value is much less than the expected recoverable reserves, the well is producing unacceptable water and remedial action should be considered if there are sufficient reserves to pay for intervention (Bailey, et al., 2000)

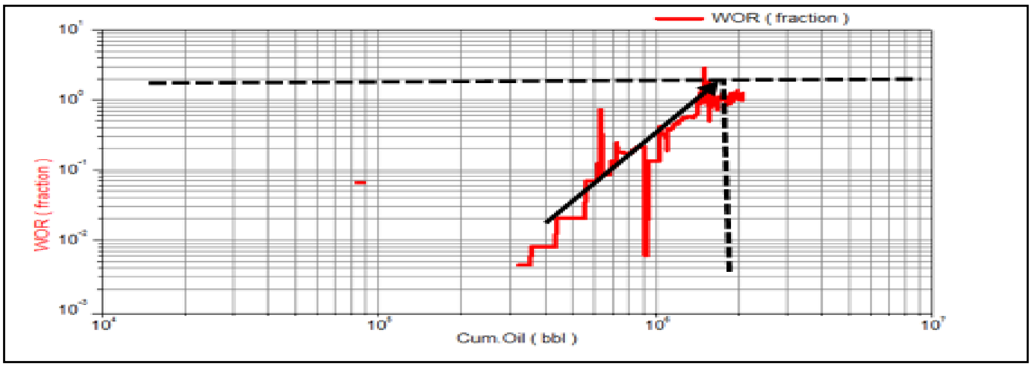


Figure 2.1: Recovery plot (Excessive Water Production Diagnosis and Strategies Analysis -Case Study- Jake Field -Sudan, 2016)

2.1.1.2 Production history plot:

This plot is a log-log plot of oil and water rates against time. Good candidates for water control usually show an increase in water production and a decrease in oil production starting at about the same time.

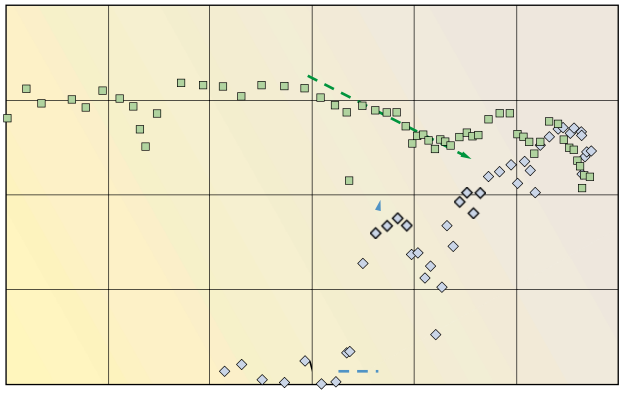


Figure 2.2: Production history plot (Bailey, et al., 2000)

2.1.2 Decline-curve analysis:

This is a semi log plot of oil production rate versus cumulative oil (below). may indicate a problem other than water, such as severe pressure depletion or damage buildup. A straight-line curve can be expected for normal depletion. An increased decline.

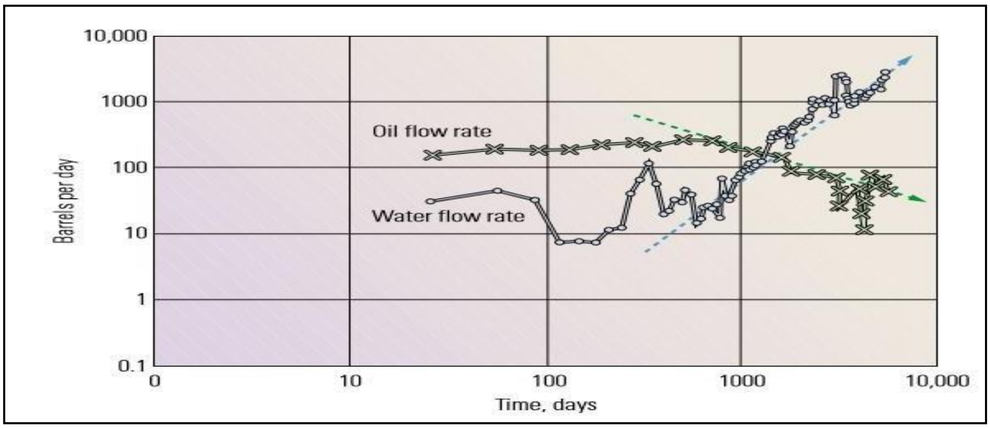


Figure 2.3: Decline curve (Bailey, et al., 2000)

2.1.3 Nodal Analysis:

suggested techniques for water production mechanism diagnosis using nodal analysis. The design of a production system depends on the combined performance of the reservoir and the downhole tubing or reservoir “plumbing” system. The amount of oil, gas and water flowing into a well from the reservoir depends on the pressure drop in the piping system, and the pressure drop in the piping system depends on the amount of each fluid flowing through it. The deliverability of a well often can be severely diminished by inadequate performance or design of just one component in the system. An analysis of a flowing wellbore and the associated piping, known as NODAL analysis, is frequently used to evaluate the effect of each component in a flowing production system from the bottom of a well to the separator. NODAL analysis is also used to determine the location of excessive flow resistance, which results in severe pressure losses in tubing systems. The effect of changing any component in the system on production rates can be determined (Bailey, et al., 2000).

2.1.4 Well Logging:

Various well logging techniques have also been developed that can be used in evaluating and predicting the water production mechanism. While they are vital tools in well and reservoir surveillance, their application during production is to some extent limiting. The logging instruments can be expensive and sometimes require shutting the well during logging which consequently affects the production rate and revenue. They could also entail costly and time-consuming log analysis and interpretation (Rabiei, et al., 2009).

2.1.5 Conventional plots:

Conventionally, water cut vs time linear plots were used to show the progress and severity of the excessive water production problems. The correlation between water cut or fractional water flow and average reservoir water saturation for two-phase flow is well known. However, it is not practical since saturation distributions throughout the reservoir are changing with time. Averaging fluid saturation from material balance does not shed any light on fluid flow behaviors in heterogeneous formations. Although these plots can also show a' drastic change in the water cut indicative of the sudden failure of well completion or rapid breakthrough of a high-water conductivity channel, the information provided by water cut plots, is limited. Regardless of multilayer channeling or coning, the shapes of the water-cut plots are very similar. Linear or semi log WOR, plots have been used to. evaluate recovery efficiency. A special plot

(Known as X-plot) that uses a correlation of. a modified fraction flow function with the recovery efficiency has also been shown to be capable of representing normal waterflood volumetric sweep efficiency. These plots could be useful to evaluate production efficiency, but they do not reveal any detail on reservoir flow behaviors. For multilayer flow, the WOR had been expressed as the ratio between the sum of the product of the permeability and the height of the water-out layers and that of the remaining oil production layer. Again, this overall estimation approach in evaluating excessive water production behavior does not shed any clue on the timing of the layer breakthrough and the relationship between the rate of change of the WOR with the excessive water production mechanism (Chan, 1995).

2.1.6 Chan’s method:

According to Chan (1995), the log-log plots of WOR (Water-Oil Ratio) versus time or GOR (Gas-Oil Ratio) versus time show different characteristic trends for different mechanisms. The time derivatives of WOR and GOR were found to be capable of differentiating whether the well is experiencing water and gas coning, high permeability layer breakthrough or near wellbore channeling. Chan identified three most noticeable water production mechanisms namely water coning, near well-bore problems and multi-layer channeling. Log-log plots of the WOR (rather than water cut) versus time were found to be more effective in identifying the production trends and problem mechanisms. it was discovered that derivatives of the WOR versus time can be used for differentiating whether the excessive water production problem as seen in a well is due to water coning or multilayer channeling. Figures (2:4) through (2:7) Chan (1995) illustrate how the diagnostic plots used to differentiate among the various water production mechanisms. Figure (2:7) shows a comparison of WOR diagnostic plots for coning and channeling. The WOR behavior for both coning and channeling is divided into three periods; the first period extends from start of production to water breakthrough, where the WOR is constant for both mechanisms. When water production begins, Chan claims that the behavior becomes very different for coning and channeling. This event denotes the beginning of the second time period. For coning, the departure time is often short (depending on several variables), and corresponds to the time when the underlying water has been drawn up to the bottom of the perforations. According to Chan, the rate of WOR increase after water breakthrough is relatively slow and gradually approaches a constant value. This occurrence is called the transition period.

For channeling, the departure time corresponds to water breakthrough for the most water-conductive layer in a multi-layer formation, and usually occurs later than for coning. Chan (1995) reported that the WOR increases relatively quickly for the channeling case, but it could slow down and enter a transition period, which is said to correspond to production depletion of the first layer. Thereafter, the WOR resumes at the same rate as before the transition period. This second departure point corresponds to water breakthrough for the layer with the second highest water conductivity. According to Chan, the transition period between each layer breakthrough may only occur if the permeability contrast between adjacent layers is greater than four. After the transition period(s), Chan describes the WOR increase to be quite rapid for both mechanisms, which indicates the beginning of the third period. The channeling WOR resumes its initial rate of increase, since all layers have been depleted. The rapid WOR increase for the coning case is explained by the well producing mainly bottom water, causing the cone to become a high-conductivity water channel where the water moves laterally towards the well. Chan (1995), therefore, classifies this behavior as channeling. Log-log plots of WOR and WOR time derivatives (WOR') versus time for the different excessive water production mechanisms are shown in Figures (2:5) through (2:7) Chan (1995) proposed that the WOR derivatives can distinguish between coning and channeling. Channeling WOR' curves should show an almost constant positive slope (Figure 2:5), as opposed to coning WOR' curves, this should show a changing negative slope (Figure 2:6). A negative slope turning positive when “channeling” occurs as shown in Figure (2:7), characterizes a combination of the two mechanisms. Chan classifies this as coning with late channeling behavior.

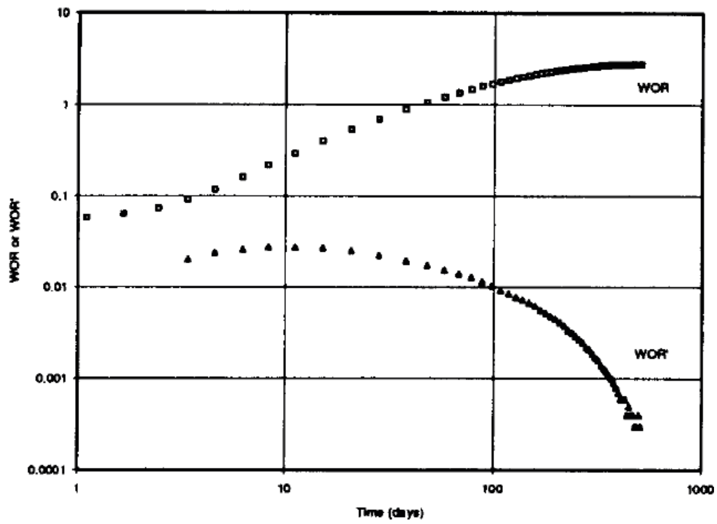


Figure 2.4: Water coning and channeling WOR comparison. Chan (1995)

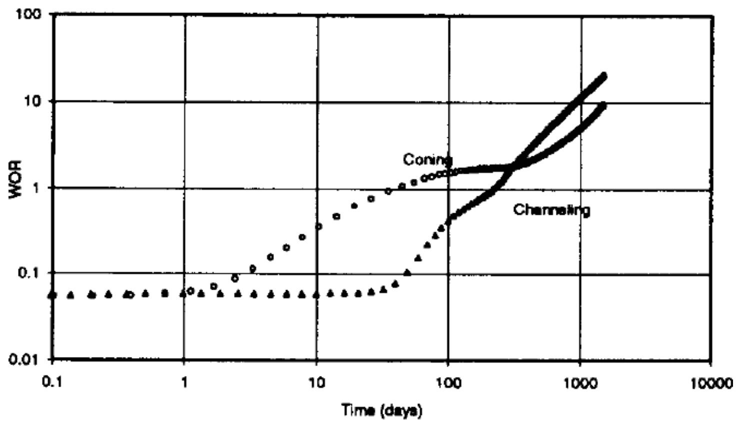


Figure 2.5: Bottom water coning with late time channeling. Chan (1995)

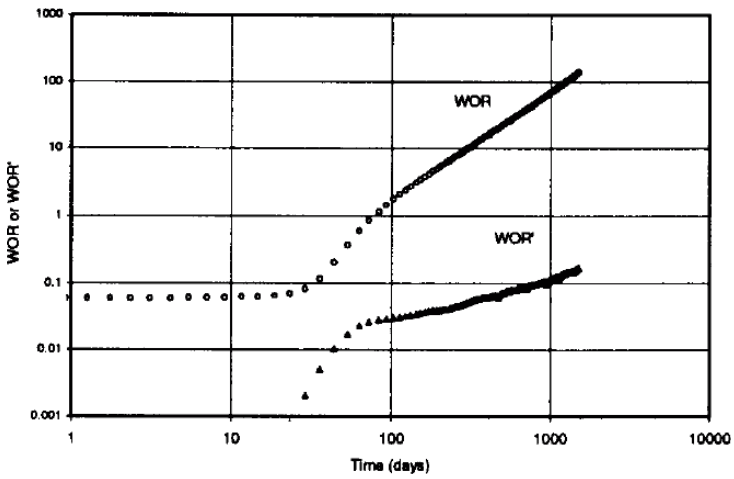


Figure 2.6: Bottom-water coning WOR and WOR'. Chan (1995)

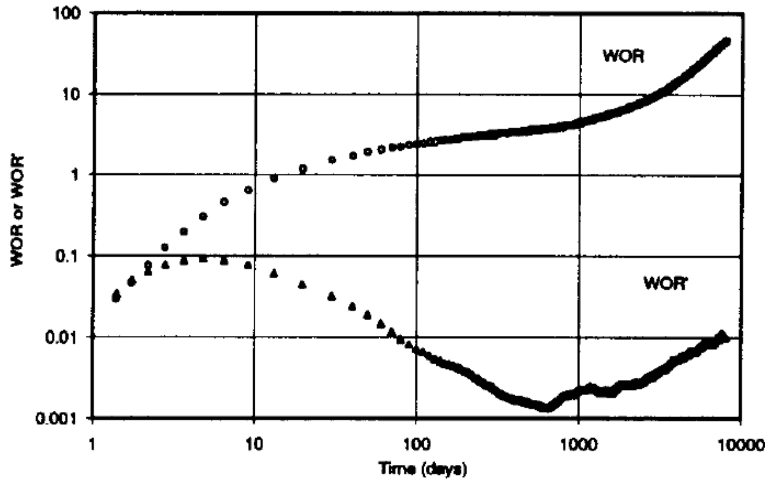


Figure 2.7: Bottom water coning with late time channeling. Chan (1995)

Recently, the use of Chan's WOR diagnostic plots has received significant interest in the oil and gas industry. However, the applications of the diagnostic plot to field data and results from numerical simulations have indicated their limitations, especially the use of derivative plots with noisy production data. There is therefore, a need to determine the validity of using these plots as a diagnostic method and to see if it can be fine-tuned (Diagnostic Plots for Analysis of Water Production and Reservoir Performance, 2010).

2.2 Machine Learning:

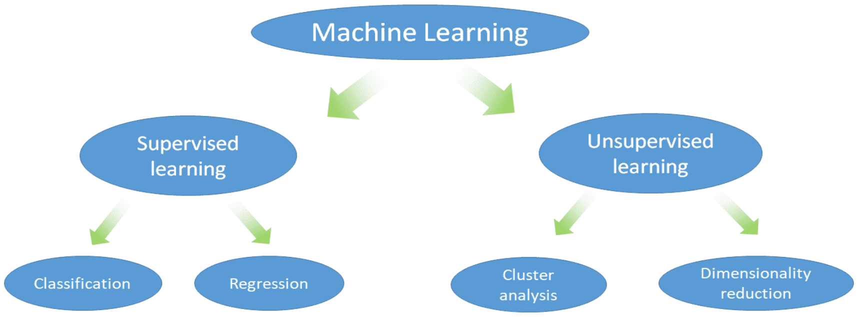


Figure 2.8: Machine learning classification (Anon., 2018)

Machine learning is about extracting knowledge from data. It is a research field at the intersection of statistics, artificial intelligence, and computer science and is also known as predictive analytics or statistical learning (Muller & Guido, 2016).

The most successful kinds of machine learning algorithms are those that automate decision-making processes by generalizing from known examples. In this setting, which is known as supervised learning, the user provides the algorithm with pairs of inputs and desired outputs, and the algorithm finds a way to produce the desired output given an input (Muller & Guido, 2016).

There are two major types of supervised machine learning problems, called classification and regression. In classification, the goal is to predict a class label, which is a choice from a predefined list of possibilities. Classification is sometimes separated into binary classification, which is the special case of distinguishing between exactly two classes, and multiclass classification, which is classification between more than two classes. You can think of binary classification as trying to answer yes/no question (Muller & Guido, 2016), our classifications in this study are coning or channeling.

2.2.1 Machine Learning Multiclass Classification Problem:

In multiclass classification problems the task is to classify instances into two or more classes; e.g., classify a well which may have patterns of constant WOR, normal displacement, multilayer channeling, and rapid channeling. Multiclass classification problem assumes that each instance is assigned one and only one label: a well can be either normal displacement or multilayer channeling but not both at the same time (Garcia, et al., 2019).

2.2.2 Ensemble classifiers:

In ensemble classification algorithms, the results from several individual classifiers are integrated in some manner (averaging or voting) in an attempt to provide a more accurate prediction (Rabiei, 2011).

It has been demonstrated through several studies in the literature that ensemble classifiers usually perform better than the individual classifiers they are based on. The classification error of a classifier can be defined by a composition of bias, variance and noise. Bias measures the difference between the predicted and actual function of the data and shows how effectively the classifier can predict the function (Rabiei, 2011).

Variance measures the variations of predictions due to changes in the learning data. Typically, there is a trade-off between bias and variance; reducing one means an increase in the other. Simple classifiers usually have low bias but high variance and complex classifiers have low variance but high bias. In ensemble classifiers, the problem of variance is taken care of by averaging the predictions from base classifiers. At the same time, given the interaction between bias and variance, ensemble classifiers can produce low biased results by using base classifiers with high variance such as classification trees (Rabiei, 2011).

To have the previous work regarding the machine learning in water production diagnostics as a guideline for our project we have found that (Rabiei, 2011) stepped away from traditional approach, extracted predictive data points from plots of WOR against the oil recovery factor. And considered three different scenarios of pre-water production, post-water production with static reservoir characteristics and post water without static reservoir characteristics for investigation. Next, they used tree-based ensemble classifiers to integrate the extracted data points with a range of basic reservoir characteristics and to unleash the predictive information hidden in the integrated data. Interpretability of the generated ensemble classifiers were improved by constructing a new dataset smeared from the original dataset, and generating a depictive tree for each ensemble using a combination of the new and original datasets. To generate the depictive tree, they used a new class of tree classifiers called logistic model tree (LMT). LMT combines the linear logistic regression with the classification algorithm to overcome the disadvantages associated with either method. Their results show high prediction accuracy rates of at least 90%, 93% and 82% for the three considered scenarios and easy to implement workflow. Adoption of this methodology would lead to accurate and timely management of water production saving oil and gas companies considerable time and money.

(Mukhanov, et al., 2018) concentrated on a mix of patterns that covers the following four production states: constant WOR, normal displacement, multilayer channeling, and rapid channeling. The former two represent acceptable production, whereas the latter two require immediate flagging and further investigation. To effectively maintain the scope of work, they deliberately excluded such water control problems as coning and thief zone formation. Moreover, both had too few examples in real data sets they used.

Moreover, they have not found by the time of this writing very specific and distinctive criteria for each water control problem pattern. Instead, there was merely a notion of linear segments and relative slope changes, without strict postulation.

On sample Chan plots of the real-life completions Figure (2:9) considered for the training or the testing of the algorithm, the X-axis reflects the number of days on production, or days elapsed from the production start date. The Y-axis reflects values for the WOR curve in blue and the WOR' curve beneath it in red.

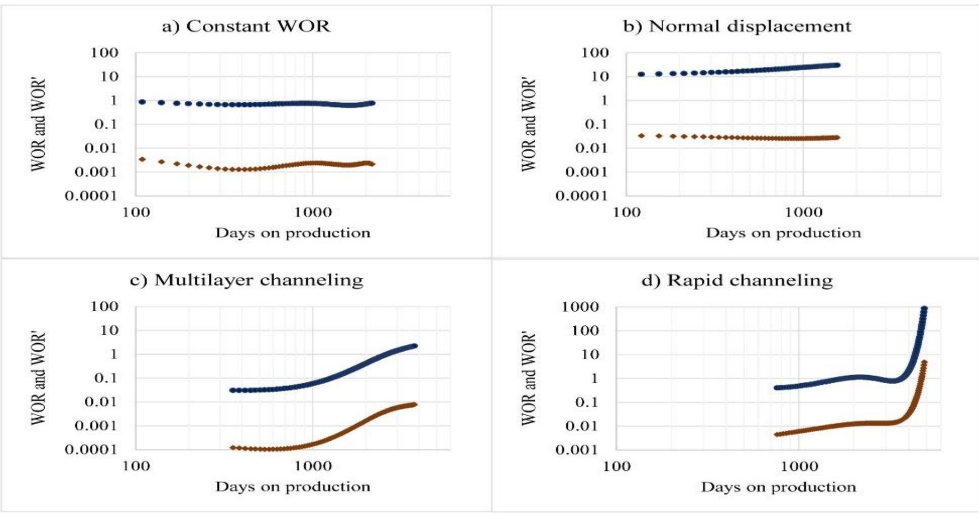


Figure 2.9: Patterns from Chan (Mukhanov, et al., 2018)

As is apparent from the figures, without a more thorough and hence more time-consuming review, in the absence of a standardized, normalized template, petroleum engineers may struggle to distinguish a visually alike yet physically different pair, such as normal displacement versus multilayer channeling (Figure (2:9) b and (2:9) c), multilayer versus rapid channeling (Figure (2:9) c and (2:9) d), and constant WOR versus normal displacement (Figure (2:9) a and (2:9) b).

Therefore, after carefully reviewing diverse real data sets, we elaborated more specific guidelines on how to distinguish among pattern types. We summarize them in Table (2:1) Magnitudes of change were assessed in terms of the number of one base ten logarithmic cycles.

ΔWOR	**ΔWOR′**	**Shape and timeframe**	**Pattern type**
< 1	1< Δ<2	Practically horizontal line throughout the most of well life	Constant WOR
~ 1	1< Δ<3	Continuous slightly upward trend often with constant slope	Normal displacement
> 1	>= 3	Growth of slopes often within half-log time cycle	Multilayer channeling
> =2	>2	Both curves rocket in a very short time of <=1/10 of log cycle	Rapid channeling

Table 2.1: Criteria elaborated for visual classification of water control patterns

An important applicability test for an ML model is whether a human expert can review features of the algorithm and confidently predict the classification output value. Therefore, having clear assumptions is crucial.

Here, we review water control diagnostic patterns in detail. For each, we will expand below on the physical meaning and the visual classification criteria which we applied to collect the training data set.

The constant WOR reflects water merely following the oil trend without accelerating contribution. This pattern represents the so-called "good" or "sweep" water that is innate or injected. When innate water is mobile in the reservoir, then it helps sweep oil. As the name suggests, the constant WOR is represented by a mostly horizontal line gravitating around single value within less than one order of magnitude. In other words, its values do not change by more than one base ten logarithm – as on Figure (2:9) a. Wells flagged as exhibiting constant WOR can be skipped from the workover candidate recognition process because they do not fall within the interest.

The normal displacement reflects another common situation in which the WOR and the WC gradually increase over time. It is natural that the WC in late-stage wells grows to as high as 80% or more; similarly, the WOR value starts off with a gradually emerging distinct upward trend. The change is consistent, characterized by an almost linear positive slope (see Figure (2:9) b). Often, normal displacement does not lift the WOR to more than one log cycle on the vertical axis. Just as wells with the constant WOR, wells demonstrating normal displacement typically do not require intervention.

The multilayer channeling refers to a sufficiently sudden and clear shift in slope from a constant WOR or normal displacement situation. The breakthrough from the most conducive layer can be traced to an initial relatively steeper and more exponential increase in the WOR in later stages of the well life, often exceeding one log cycle on the vertical axis (see Fig. 1c). On the horizontal axis, this pattern type often develops within a half-log timeframe. A stair-step like shape reflects transition periods and layered-permeability contrasts, as breakthroughs from other nearby layers intensify the slope increase. The pattern represents edge-water flow or a moving oil-water contact (OWC).

The rapid channeling is characterized by an abrupt and quickly intensifying water problem or a barrier breakdown, such as a near wellbore breakthrough, an open flow path through a fault, a fracture, or a channel behind casing in the casing-formation annulus due to a poorly implemented cement job. Note a very significant change of WOR in a short period of time on Fig. 1d. We considered as a strong visual cue for the rapid channeling pattern a WOR change of two or more orders of magnitude within the late-time period. In the presented example, an increase by three log cycles, i.e., a WOR increase from 1 to 1000, is observed in approximately 20% of the latest portion of the well life within less than or equal to one-tenth of a late time log cycle.

Chapter (3)

Methodology

3.1 Introduction:

In an integrated methodology a machine learning model have been built with python programming language libraries using ridge classifier algorithm. This model predicts the excessive water production mechanism, and it is result have been compared with conventional investigation (human decision) which ensure that the model is providing an accurate result.

3.2 General Procedure:

1. Data Collection.
2. Data cleaning and preparation.
3. Data classification (Coning, Channeling).
4. Creating an ensemble of several models each model has data with relatively close trends to achieve overfitting,
5. Passing unseen data to the ensemble to make sure it is able to generalize from the training data to a new data.

3.3 Why machine learning?

1. Lack of sufficient human expertise in a domain.
2. Scenarios and behavior can keep changing over time.
3. Human have sufficient expertise in the domain but it is extremely difficult to formally explain or translate this expertise into computational task.
4. Addressing domain specific problems at scale with huge volume of data with too many complex conditions and constrain.

3.4 why python?

Python has become the lingua franca for many data science applications. It combines the power of general-purpose programming languages with the ease of use of domain-specific scripting languages like MATLAB or R. Python has libraries for data loading, visualization, statistics, natural language processing, image processing, and more. This vast toolbox provides data scientists with a large array of general- and special-purpose functionality. One of the main advantages of using Python is the ability to interact directly with the code, using a terminal or other tools like the Jupyter Notebook. Machine learning and data analysis are fundamentally iterative processes, in which the data drives the analysis. It is essential for these processes to have tools that allow quick iteration and easy interaction. As a general-purpose programming language, Python also allows for the creation of complex graphical user interfaces (GUIs) and web services, and for integration into existing systems (Muller & Guido, 2016).

3.5 Concept of Ridge Classifier:

The Ridge Classifier, based on Ridge regression method, converts the label data into [-1, 1] and solves the problem with regression method. The highest value in prediction is accepted as a target class and for multiclass data multi-output regression is applied.

Ridge regression is a linear regression regularized with L2 norm. The matrix formulation is shown in Equation (3:1), which is obtained from minimizing the sum of squares of residuals (Y and Yˆ).

$$\boldsymbol{W} = (\boldsymbol{X^T X} + \propto \boldsymbol{I})^{-1} \boldsymbol{X^T Y} \text{...} \textbf{ Equation (3:1)}$$

X is the input matrix of size m x n, where m is the number of data points and n is the total number of features. Y is the output vector size m. W is the vector of parameters (including bias term). $\propto$ is a hyperparameter that controls regularization strength. We need to adjust regularization strength such that the model is able to accurately model the training set (or minimize the bias) as well as perform well in other datasets with unseen inputs (or minimize the variance). This is a bias-variance trade-off problem.

Trying to fit every single data point in the training set can lead to high variance or overfitting. The high variance can be reduced by penalizing W with ∝ as W grows larger. In other words, ∝ shrinks the contribution of each feature in X.

In the training period, given a pair of predictor X, ground truth Y, and user-defined ∝, we can obtain W. In this research, optimum ∝ was obtained through hyperparameter grid search with ∝ = [0.01, 1, 1, 2, 5, 10, 100, 1000]. After obtaining W and ∝, the prediction can be calculated using Equation (3:2) (Ristanto, 2018)

$$\boldsymbol{Y = WX} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \textbf{Equation (3:2)}$$

3.6 Concept of Ensemble Classifier:

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote (Avik_Dutta, 2022).

3.7 Used Python Libraries:

3.7.1 NumPy:

NumPy, short for Numerical Python, has long been a cornerstone of numerical computing in Python. It provides the data structures, algorithms, and library glue needed for most scientific applications involving numerical data in Python. NumPy contains, among other things:

• A fast and efficient multidimensional array object ndarray

• Functions for performing element-wise computations with arrays or mathematical operations between arrays • Tools for reading and writing array-based datasets to disk

• Linear algebra operations, Fourier transform, and random number generation (McKinney, 2017).

3.7.2 Pandas:

pandas provides high-level data structures and functions designed to make working with structured or tabular data fast, easy, and expressive. Since its emergence in 2010, it has helped enable Python to be a powerful and productive data analysis environment. The primary objects in pandas that will be used in this book are the DataFrame, a tabular, column-oriented data structure with both row and column labels, and the Series, a one-dimensional labeled array object. Pandas blends the high-performance, array-computing ideas of NumPy with the flexible data manipulation capabilities of spreadsheets and relational databases (such as SQL). It provides sophisticated indexing functionality to make it easy to reshape, slice and dice, perform aggregations, and select subsets of data. Since data manipulation (McKinney, 2017).

3.7.3 Scikit-learn:

scikit-learn contains a number of state-of-the-art machine learning algorithms, as well as comprehensive documentation about each algorithm. Scikit-learn is a very popular tool, and the most prominent Python library for machine learning. It is widely used in industry and academia.

It includes submodules for such models as:

• Classification: ridge classifier, SVM, nearest neighbors, random forest, logistic regression, etc.

• Regression: Lasso, ridge regression, etc.

• Clustering: k-means, spectral clustering, etc.

• Dimensionality reduction: PCA feature selection, matrix factorization, etc.

• Model selection: Grid search, cross-validation, metrics

• Preprocessing: Feature extraction, normalization Along with pandas. Scikit-learn has been critical for enabling Python to be a productive data science programming language (McKinney, 2017).

3.8 Coding Procedure:

To build a machine learning model there are major steps, first step is data preprocessing (data cleaning and scaling), it is common practices is to adjust the features so that the data representation is more suitable to algorithm, second step is modeling and the third step is model evaluation

3.8.1 Data cleaning:

In data cleaning and preparation three main functions, create_csv_files(), prep_well() and remove_before_production() are defined inside each there are many functions that do a specific task.

1. Pandas and NumPy libraries are imported and then RidgeClassifier algorithm is imported from sklearn library:

```
[1]: import pandas as pd
     import numpy as np
     from sklearn.linear_model import RidgeClassifier
```

2. The data is in form of an excel file, this file includes a sheet for each well, the desired wells to be used are stored in a list variable named sheets and then saved in a file by np.save().

```
[2]: np.save("sheets", sheets)
```

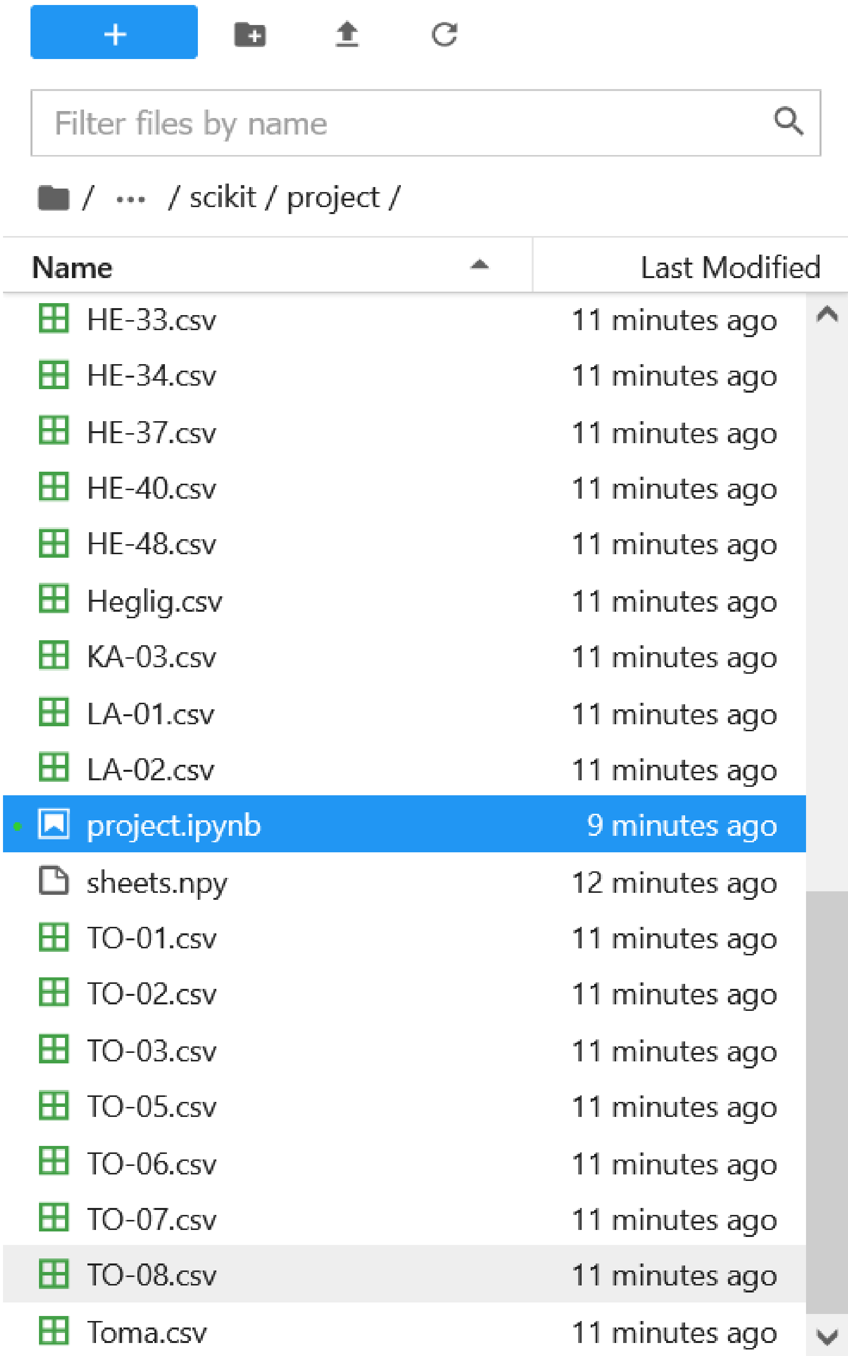
3. create_csv_files() is defined which takes excel file name as an argument then desired wells names are loaded from sheets file then creates a csv for each well.

```
[3]: def create_csv_files(excel_file_name):

         sheets = np.load("sheets.npy")

         for sheet in sheets:
             read_file = pd.read_excel("../excel/" + excel_file_name, sheet_name = sheet, header = 1)
             read_file.to_csv(sheet+".csv", index = 0,header=True)
```

```
[4]: create_csv_files("Greater_Heglig.xlsx")
```



4. Now each well has a file. To make sure that everything is ok and data columns are exist and in the top of the sheet get_well_info() is defined which takes the file name and reads the file using read_csv(), if columns are existed in the top of the sheet it returns a dictionary, the first key in the dictionary is ready status it's value is True, the second key contains the dataframe. If the columns are not existed in the top of the sheet, then it returns a dictionary, the first key is ready status which is False, the second key contains the dataframe and the third key is the true index of columns.

```
[5]: def get_well_info(well_name):
         make_df = pd.read_csv(well_name + ".csv" )

         try:
             make_df['WOR']
             ready = True
             result = {
                 "ready" : ready,
                 "df" : make_df
             }

         except(KeyError):
             get_index = make_df[make_df['      START_DATE '] == "DATE"].index.values
             ready = False
             result = {
                 "ready" : ready,
                 "df" : make_df,
                 "index" : get_index
             }
         return result
```

5. prep_well() passes the file name to get_well_info() and do certain operations depend on it's result, fill NaN values with 0 and eventually returns a dataframe consists of two columns [WOR, WOR'].

```
[6]: def prep_well(well_name):
         well_info = get_well_info(well_name)
         df = well_info["df"]
         is_ready = well_info["ready"]

         if is_ready:
             df = df.drop(df.index[0],axis = 0).reset_index()

         else:
             df.drop(df.index[0:int(well_info["index"])], axis = 0,inplace = True)
             df.columns = df.iloc[0]
             df = df.drop(df.index[[0,1]],axis = 0).reset_index()

         df = df[["WOR","WOR'"]].fillna(0)
         return df
```

```
[7]: prep_well("LA-01")
```

158	WOR	WOR'
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...	...	...
148	35.06201550387597	0
149	10.353280263519078	0
150	17.17577120822622	0.22008035305506907
151	16.691418137553256	0
152	0	0

153 rows × 2 columns

6. Some field's wells have started production late than other wells and the period before the start of production was" filled "with zero value which makes problems in departure time as one of the classification parameters

so before production period problem is solved by removing the zero values from WOR column.

```
[8]: def remove_before_production(well_name):

         n = 0

         for i in range(0, len(df["WOR"])):
             if df["WOR"].iloc[i] == "0":
                 n = n+1
             else:
                 break

         df =  df.drop(df.index[0 : n], axis = 0).reset_index()

         return df
```

7. make_wells_ready() load the list of sheets from sheets.npy file , loops on it and passes each well to prep_well() function to return the ready DataFrame. Then remove zero values before production.
60 features for each well has been considered as it's the least number of features for a well in (training and test) data so that model will be built by wells that has complete resemblance.

```
[9]: def make_wells_ready():

    sheets = np.load("sheets.npy")

    for sheet in sheets:
        df = prep_well(sheet)
        df = remove_before_production(df)
        df = df.iloc[0:60]
        df.to_csv(sheet+".csv", index = 0,header=True)
```

3.8.2 Modeling:

Cell below is the function used to read files

```
[10]: def read_well(well_name):

    df = pd.read_csv(well_name + ".csv")

    return df
```

1. The WPM depends on trends of WOR and WOR’, so each one is a separate datapoint. make_model_dataset() takes a dictionary of wells names as a keys and classes (0 for coning and 1 for channeling) as values and from the dictionary make lists for WOR, WOR’ and classes and return them.

```
[11]: def make_model_dataset(wells_name_class):

    wor = []
    wor_deriv = []
    classes = []

    wells = wells_name_class.keys()
    labels = wells_name_class.values()

    for well in wells:
        df = read_well(well)
        wor.append(df["WOR"])
        wor_deriv.append(df["WOR'"])


    for label in labels:
        classes.append(label)

    return {
        "wor" : np.array(wor),
        "wor_deriv" : np.array(wor_deriv),
        "classes" : np.array(classes),
    }
```

2. Next function is used to build (instantiate) the machine learning model with ridge classifier algorithm. RidgeClassifier() encapsulate the algorithm that used to build the model as well the algorithm to make predictions on new datapoints. To train the model, fit method is called which takes as arguments train data (wor or derivative) and its corresponding labels.

```
[12]: def model_instantiate(wells_name_class):

          data = make_model_dataset(wells_name_class)

          wor_classifier   = RidgeClassifier().fit(data["wor"], data["classes"])
          deriv_classifier = RidgeClassifier(alpha = 0.1).fit(data["wor_deriv"], data["classes"])

          return { "wor_classifier" : wor_classifier,
                   "deriv_classifier" : deriv_classifier}
```

3. model_overfit() function instantiate a model by data and measure overfitting by passing the same data to score() function.

```
[13]: def model_overfit(wells_name_class):

          # data
          data = make_model_dataset(wells_name_class)

          # classifiers
          classifiers = model_instantiate(wells_name_class)

          wor_classifier   = classifiers["wor_classifier"]
          deriv_classifier = classifiers["deriv_classifier"]

          wor_overfitting   = wor_classifier.score(data["wor"], data["classes"])

          deriv_overfitting = deriv_classifier.score(data["wor_deriv"], data["classes"])

          return {
              "wor_overfitting" : wor_overfitting,
              "deriv_overfitting" : deriv_overfitting
          }
```

4. Model_genralize() measure the generalization by passing test data (new data) to score() function.

```python
[14]: def model_generalize(wells_train_data, test_data) :

    # test data
    test_data = make_model_dataset(test_data)

    # classifiers
    classifiers = model_instantiate(wells_train_data)

    wor_classifier   = classifiers["wor_classifier"]
    deriv_classifier = classifiers["deriv_classifier"]

    wor_score   = wor_classifier.score(test_data["wor"], test_data["classes"])

    deriv_score = deriv_classifier.score(test_data["wor_deriv"], test_data["classes"])

    return {
        "wor_score"   : wor_score,
        "deriv_score" : deriv_score
    }
```

5. To build the final model which combine several models for each trend next function is defined which loads a list of trends from trends.npy.

```python
[15]: def ensemble_classifier():

    trends = np.load("trends.npy", allow_pickle=True)

    models = ["model"] * len(trends)

    for i, trend in enumerate(trends):
        models[i] = model_instantiate(trend)

    np.save("ensemble_classifier", models)
```

3.8.3 Model Evaluation:

prep_new-data() takes a well name and return WOR and WOR'.

```python
[16]: def prep_new_data(well_name):

    df = read_well(well_name)

    wor = df["WOR"]
    deriv = df["WOR'"]

    return {
        "wor" : wor,
        "deriv" : deriv
    }
```

1. predict function is used to make predictions from models and add the votes for WOR and WOR' in a list.

```python
[17]: def predict(well_name):

    models = np.load("ensemble_classifier.npy", allow_pickle=True)

    predictions = []

    data = prep_new_data(well_name)
    wor = data["wor"]
    deriv = data["deriv"]

    for model in models:

        predictions.append(int(model["wor_classifier"].predict([wor])))
        predictions.append(int(model["deriv_classifier"].predict([deriv])))

    predictions = np.array(predictions)

    return predictions
```

2.From the predict function the decide() function takes the final decision whether the well is coning or channeling (0,1).

```python
[18]: def decide(well_name):

    predictions =  predict(well_name)

    channeling = len(predictions[predictions == 1])
    coning = len(predictions[predictions == 0])

    if channeling > coning:
        decision = 1
    else:
        decision = 0


    return decision
```

3. ensemble_score() is used to compute the accuracy of the ensemble classifier by make predictions for each well in the test data and compare it against it is label (the known mechanism).

[19]:
```python
def ensemble_score(test):

    decisions = []
    classes = []

    wells = test.keys()
    labels = test.values()

    for label in labels:
         classes.append(label)

    for well in wells:
        decision = decide(well)
        decisions.append(decision)


    decisions = np.array(decisions)
    classes   = np.array(classes)
    score = np.mean(decisions == classes)

    print ("ensemble score : " + str(score * 100) + "%")
```

Chapter 4

Results and Discussion

In this chapter, we present the results of ensemble classifier used for identifying two different WPMs. Two thirds of the cases in the data are used for building and training the ensemble classifier. The remaining cases are used for evaluating the efficiency of the final model.

To choose the suitable trends to train the models, frequent trial and error executed in the training data to achieve balance between overfitting and generalization. In data science and machine learning Overfitting is a common undesirable concept in the model performance, happens when the model learns details (overfit) in training and cannot perform accurately against unseen data, but in prediction of WPMs the trajectory of WOR and WOR' is traced so there is a need to overfit the curve.

```
[21]: model_overfit(trend1)
[21]: {'wor_overfitting': 1.0, 'deriv_overfitting': 0.75}
```

```
[22]: model_overfit(trend2)
[22]: {'wor_overfitting': 1.0, 'deriv_overfitting': 0.8571428571428571}
```

```
[23]: model_generalize(trend1, test)
[23]: {'wor_score': 1.0, 'deriv_score': 0.6666666666666666}
```

```
[24]: model_generalize(trend2, test)
[24]: {'wor_score': 1.0, 'deriv_score': 0.5555555555555556}
```

The table below shows the chosen trends for constructing the ensemble classifier beside the test data.

Train set		Test set	Classifications
Trend1	**Trend2**		
"LA-01"	"EF-02"		Channeling
"BA-01"	"HE-06"	"TO-02"	
"EB-01"	"HE-10"	"HE-05"	
"EF-02"	"HE-13"	"HA-02"	
"El Bakh"	"HE-14"	"Heglig"	
"El Full"	"HE-17"		
"TO-01"	"Toma"		
"TO-03"			
"EF-04"	"HE-26"	"HA-03"	coning
"HE-34"	"HE-34"	"HE-15"	
"HE-26"	"LA-02"	"HE-33"	
"LA-02"	"TO-07"	"HE-37"	
"TO-07"	"BA-04"	"KA-03"	
"TO-08"			

70% ⟷ 30%

Table 4.1: Trend set and Test set

The generalization without applying any restriction for both models in WOR was 100% because it is distinguished by three different periods (departure time, transition period and third period), but for WOR' the two models' generalization is 0.67 and 0.56 respectively because it has only one indicator (positive or negative slope) and there a considerable share zone in the slope whether it is positive or negative.

We studied the efficiency of decreasing alpha parameter in overfitting and generalization for WOR' and we ended up with this result:

Trend (1)

Alpha parameter	overfitting	generalization
0	75%	67%
0.1	100%	89%
0.01	100%	89%
0.001	100%	100%
0.0001	100%	100%

Table 4.2: alpha parameter effect on model (1)alpha parameter effect on model (1)

Trend (2)

Alpha parameter	overfitting	generalization
0	86%	56%
0.1	100%	78%
0.01	100%	78%
0.001	100%	89%
0.0001	100%	89%

Table 4.3: alpha parameter effect on model (2)

These trends are listed in a variable named trends and then saved in a file

```
[41]: np.save("trends", trends)
```

By calling ensemble_classifier function final model will be built and saved in a file named ensemble_classifier.npy

```
[42]: ensemble_classifier()
```

Taking a decision from one model has a problem, sometimes one model would give coning from one classifier and channeling from another, in this case model will fail to decide what is the exact mechanism. to solve this problem, we built an (ensemble classifier)

The ensemble classifier improves the accuracy, it is like having a number of experts discussing a particular issue and voting on a decision, for example "Hegilg" is a well producing water by channeling mechanism:

```
[81]: model1 = model_instantiate(trend1)
      model2 = model_instantiate(trend2)
```

```
[82]: print(model1["wor_classifier"].predict([well_data["wor"]]) ,model1["deriv_classifier"].predict([well_data["deriv"]]))
      [1] [1]
```

Model1 predicts channeling mechanism[1] from WOR classifier and WOR’ classifier.

```
[83]: print(model2["wor_classifier"].predict([well_data["wor"]]) ,model2["deriv_classifier"].predict([well_data["deriv"]]))
      [1] [0]
```

But model2 predicts channeling mechanism [1] from WOR classifier and coning mechanism [0] from WOR’ classifier. In this case it is impossible to take a decision, and we can notice the failure of model1.

```
[78]: predict("Heglig")
[78]: array([1, 1, 1, 0])
[80]: decide("Heglig")
[80]: 1
```

The final decision of the ensemble classifier is that the well is producing

water by channeling mechanism [1]

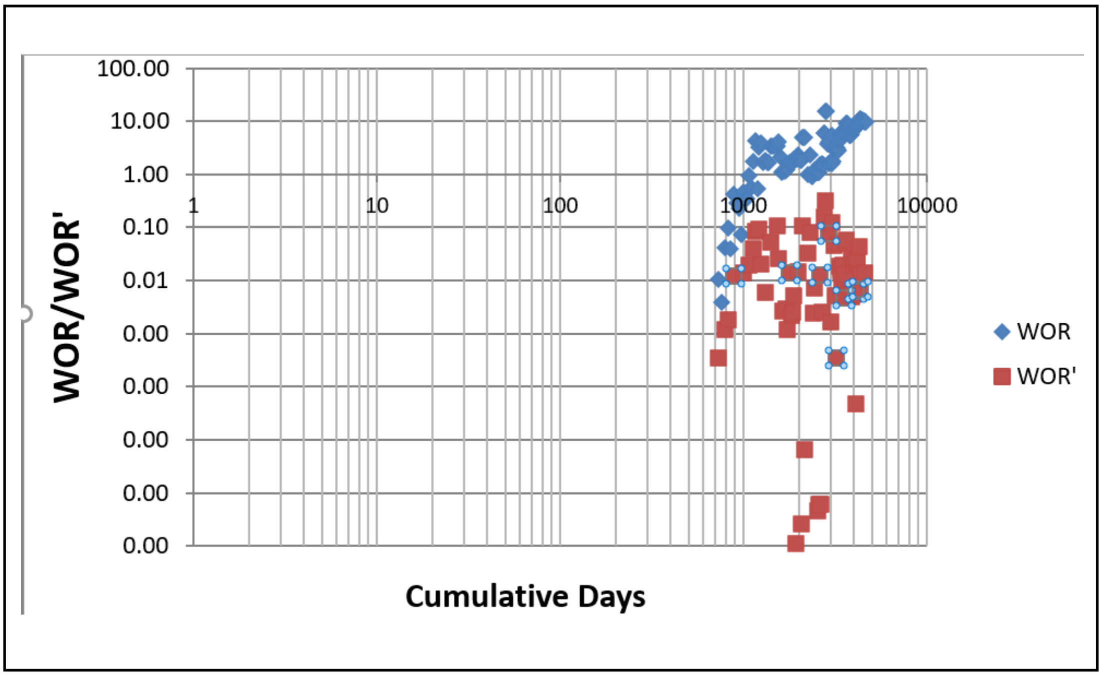


Figure 5.1: KA-03 Diagnostic plot

From conventional investigation KA-03 well trend for WOR show that there is very short departure time which is a strong indicator for coning, after breakthrough (second period) the WOR increase relatively quickly near to be channeling, and it is gradually approaches constant value (coning), after transition period WOR is not resuming it's initial rate of increase that is an indicator for coning. In the derivative curve there is uncertainty whether it is coning or channeling but it is seemed to be coning if we considered the hall trend.

The final decision for the production mechanism from the conventional investigation is coning.

The class from the Ridge Classifier ML model: is 100% coning

```
[26]: predict("KA-03")
[26]: array([0, 0, 0, 0])
```

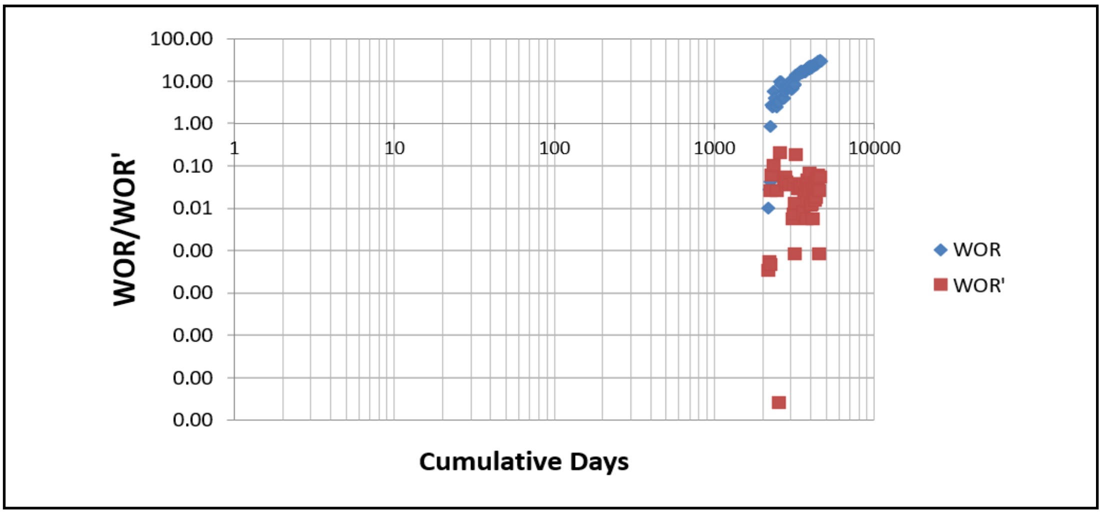


Figure 4.2: HE-33 Diagnostic plot

From conventional investigation HE-33 well trend for WOR show that there is very short departure time and after breakthrough (second period) the WOR increase slowly which are strong indicators for coning. In the derivative curve the negative slope clearly shows that it's coning.

The final decision for the production mechanism from the conventional investigation is coning.

The class from the Ridge Classifier ML model: is 100% coning

```
[29]: predict("HE-33")
[29]: array([0, 0, 0, 0])
```

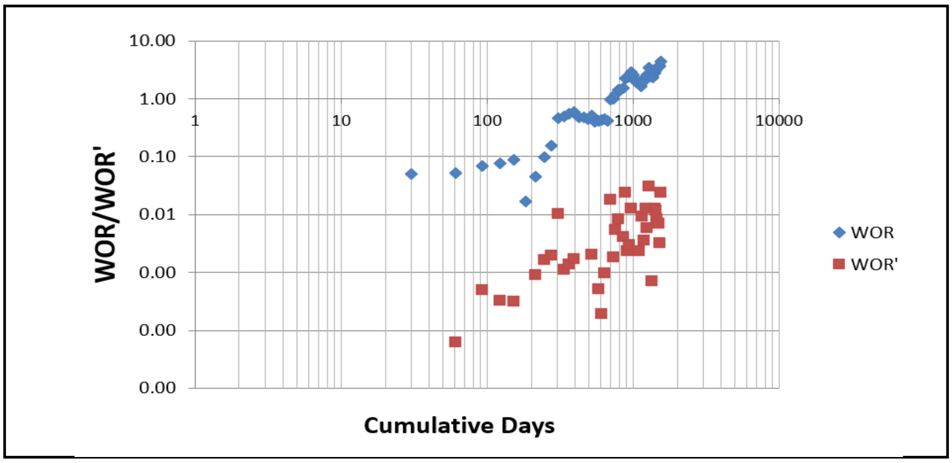


Figure 4.3: Heglig Diagnostic plot

From conventional investigation Hegilg well trend for WOR show that there is long departure time which is a strong indicator for channeling, after breakthrough (second period) the WOR increase relatively slowly near to be coning, and it is gradually approaches constant value (coning), after transition period WOR is resuming it's initial rate of increase that is an indicator for channeling. In the derivative curve the positive slope is indicator for channeling.

The final decision for the production mechanism from the conventional investigation is channeling.

The class from the Ridge Classifier ML model : is 75% channeling

```
[30]: predict("Heglig")

[30]: array([1, 1, 1, 0])
```

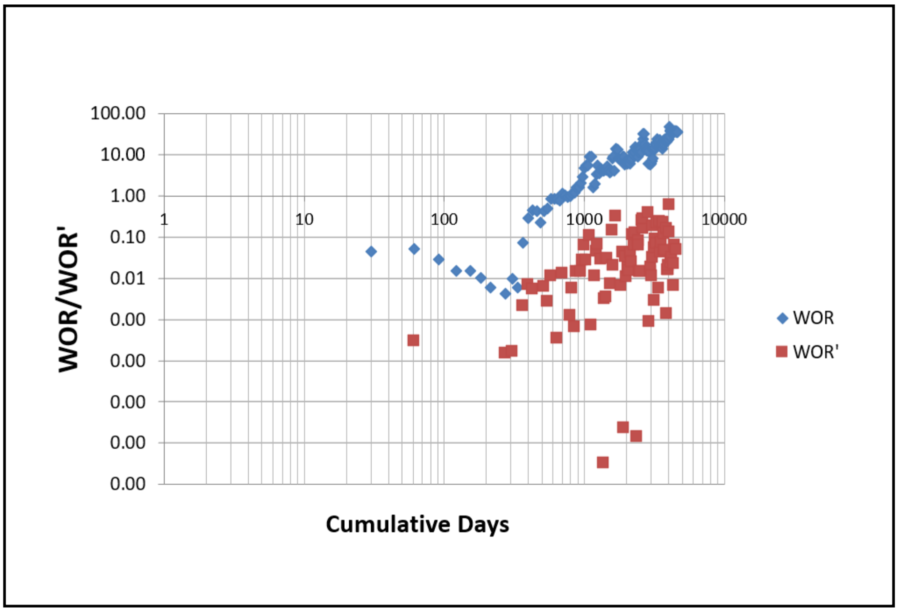


Figure 4.4: TO-02 Diagnostic plot

From conventional investigation TO-02 well trend for WOR show that there is very long departure time and after breakthrough (second period) the WOR increase relatively quickly which are strong indicators for channeling. In the derivative curve the positive slope is indicator for channeling.

The final decision for the production mechanism from the conventional investigation is channeling.

The class from the Ridge Classifier ML model: is 100% channeling

```
[30]: predict("TO-02")
[30]: array([1, 1, 1, 1])
```

The ensemble classifier predicts the remaining test set as follows:

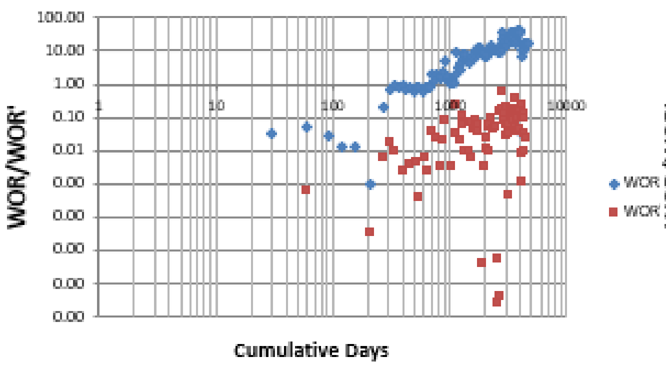


Figure 4.5: HA-02 Diagnostic plot (channeling)

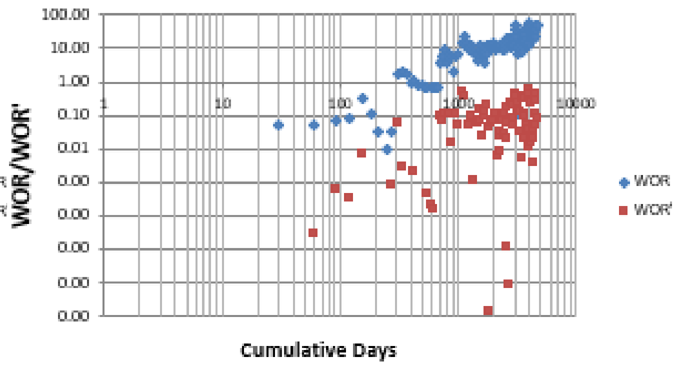


Figure 4.6: Figure 18 HE-05 Diagnostic plot (channeling)

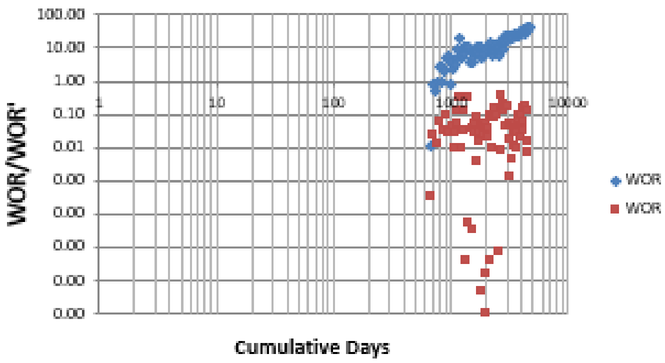


Figure 4.7: Figure 18 HA-03 Diagnostic plot (coning)

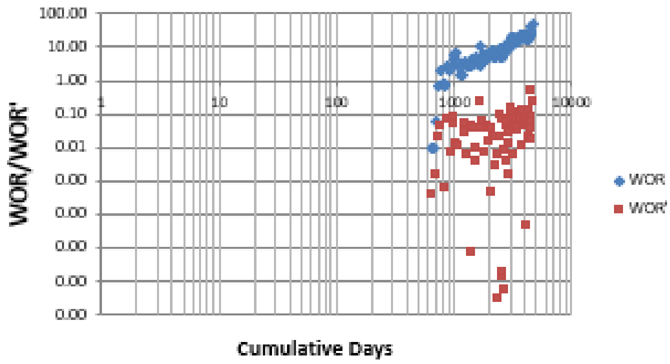


Figure 4.8: Figure 18 HE-15 Diagnostic plot (coning)

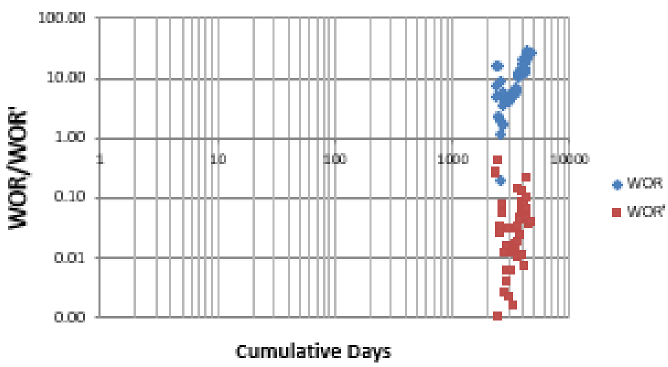


Figure 4.9: Figure 18 HE-37 Diagnostic plot (coning)

Comparing test wells classes (already known) to the predictions of the ensemble we found that the model gives 100% accuracy.

```
[84]: ensemble_score(test)

      ensemble score : 100.0%
```

Chapter 5

Conclusions & Recommendations

5.1 Conclusions:

1- In this work we presented a machine learning model based on Chan diagnostic plot pattern using Ridge classifier, we started our work by cleaning and preparing the given data. we identified the water production mechanisms for 30 wells split into 21wells for training and 9 wells for testing. Two trends have been considered from the train data to build the models.

2- We studied the effectiveness of decreasing the hyperparameter in achieving highest overfitting with the best generalization, and it turns out to be $\alpha = 0.001$. we combined the two models in one ensemble classifier to increase the accuracy of predictions. Finally, after passing the test data to the ensemble classifier model could achieve score of 100% on the test set.

5.2 Recommendations

1. The number of models should be increased to boost the decision boundary of the ensemble classifier.

2. Water cut slope could be included as third classifier.

3.Additional models could be trained for specific pattern from Chan plots (constant WOR, normal displacement, multilayer channeling, rabid channeling …etc.

References

1. Okon, A. N., Appah, D. & Akpabio, J. . U., 2017. Water Coning Prediction Review and Control:Developing an Integrated Approach. *Journal of Scientific Research & Reports.*
2. Anon.,2018.*DiegoCalvo.*[Online] Availableat:https://www.diegocalvo.es/en/author/diegocalvo/ [Accessed 2021].
3. Avik_Dutta,2022.*GeeksforGeek.*[Online] Availableat:https://www.geeksforgeeks.org/ensemble-classifier-data-mining/ [Accessed 2 2022].
4. Bailey, B. M. C. J. T. et al., 2000. Water Control. *Schlumberger,* p. 37.
5. Chan, K., 1995. *Water Control Diagnostic Plots.* Dallas, s.n.
6. *Diagnostic Plots for Analysis of Water Production and Reservoir Performance* (2010) Echufu-Agbo Ogbene Alexis.
7. *Excessive Water Production Diagnosis and Strategies Analysis - Case Study- Jake Field -Sudan* (2016) Mohanned Mahgoup Mohammed Khiary.
8. Garcia, C. A., Mukhanov, A. & Torres, H., 2019. *Chan Plot Signature Identification as a Practical Machine Learning Classification Problem.* Beijing, s.n.

9. McKinney, W., 2017. *Python for Data Analysis.* Second Edition ed. Sebastopol: O’Reilly Media.

10. Moradi, B., Dastkhan, Z., Roozbehani, B. & Montazeri, G., 2010. *Modeling of water coning phenomena in Fractured reservoir and design a simulator.* Trinidad and Tobago, s.n.

11. Mukhanov, A., Garcia, C. A. & Torres, H., 2018. *Water Control Diagnostic Plot Pattern Recognition Using Support Vector machine.* Moscow, s.n.

12. Muller, A. C. & Guido, S., 2016. *Introduction to machine learning with python.* First ed. Sebastopol: O'Reilly media.

13. Rabiei, M., 2011. *Excess Water Production Diagnosis in Oil Fields Using Ensemble Classifiers.* Perth: s.n.

14. Rabiei, M., Gupta, R., Cheong, Y. P. & Soto, G. A. S., 2009. *Excess Water Production Diagnosis in Oil Fields using Ensemble Classifiers.* Perth: s.n.

15. Ristanto, T., 2018. *MACHINE LEARNING APPLIED TO MULTIPHASE PRODUCTION PROBLEMS.* STANFORD: s.n.

16. Taha, A. & Amani, . M., 2019. Overview of Water Shutoff Operations in Oil and Gas Wells; Chemical and Mechanical Solutions. *ChemEngineering,* p. 1.