University of
Stavanger

Faculty of Science and Technology

MASTER’S THESIS

Study program/ Specialization: Petroleum Engineering – Well Engineering	Spring semester, 2019 Open access
Writer: Jakub Frankiewicz	………………………………………… (Writer’s signature)

Faculty supervisor:
Dan Sui

Faculty co-supervisor:
Ekaterina Wiktorski

External supervisor(s):

Thesis title:

The Application of Data Analytics and Machine Learning for Formation Classification and Bit Dull Grading Prediction

Credits (ECTS): 30

Key words: Bit Dull Grading Formation Classification Data Analytics Machine Learning Python	Pages: 76 + enclosure: 13 Stavanger, 15.06.2019

Abstract

The oil and gas industry, especially its upstream part generates a massive amount of data. The proper data collection and processing are the vital elements of reducing the non-productive time and increasing the drilling operations efficiency.

The major part of each well program is the drill bits selection. It is the most important tool which does slicing or crushing downhole and highly affects the overall drilling performance. However, drill bit selection is mostly accomplished through lessons learned from previous runs as well as bit grading after each run. These methods are highly subjective and usually based on the engineer's experience.

The abundance of field data with data analytics and machine learning capabilities are a perfect combination for creating reliable data-driven models. The main objective of this study is to create robust models that are able to classify the formation based on drilling parameters as well as estimate the bit dull grading based on drilling parameters and the formation. In order to achieve the aforementioned goals, the disclosed Volve filed dataset was meticulously processed and analyzed.

The models were created for each of the well sections by using the Python, especially the pandas and scikit-learn libraries. However, after running the first simulation, models usually showed unsatisfactory accuracy. In order to increase models performance, the code was written to find the best parameter for each machine learning technique. Even though the bit dull grading model has a valid algorithm, the input parameters are hard to find, due to the lack of literature and patterns.

Obtained results proved that the machine learning technique may be successfully implemented to solve the everyday problems in the oil and gas industry. Moreover, the outcome should help in the well planning process, enables to decrease the number of trips and improves overall drilling phase efficiency. The process could eliminate the trial and error drill bits selection and ensure more efficient and effective decision-making process.

Table of contents

1. Thesis Introduction and Objectives

Nowadays, the role of data is significantly increased. Understanding the possessed data may lead to gaining the technical and technological advantage over competitors. In such a demanding environment as oil and gas industry information plays a key role between finding the new oil field or drilling another dry hole and counting losses.

The amount of produced data by each well is enormous and it is hardly possible for a human being to be able to read it quickly and draw proper conclusions. This is the reason why the data-driven approach becomes more and more popular not only in the oil and gas industry but within any sector which deals with abundant datasets. Such an approach, when used properly, may cut the time for obtaining valuable information form possessed data and may give results which help the companies to cut costs and improve the profits.

The work in the thesis is based on the Volve field dataset which was disclosed in June 2018 by Equinor. The dataset contains a wide spectrum of information, but this work takes only into account the drilling and logging data. Hence, the thesis can be divided into two separate cases.

The first case is the formation classification based on the drilling data. In this part, based on the prepared datasets the classification machine learning algorithms have been used to predict the formation. However, due to the varied lithology, only the datasets with the well sections 12 ¼” and 8 ½” were chosen to be input for the model.

The second case is the bit dull grading prediction. In this part, there is no labelled data and the regression machine learning algorithms were used to predict the bit wear. While working on this part, it was discovered that despite the lack of literature on the subject, currently, the major service companies work on finding the solution on how to predict the bit wear accurately. It is the burning issue because the drill bit is one of the key components of the drilling process which interacts with the formation and so far not much information are collected about the bit state while drilling. The datasets for this problem included well sections 26”, 17 ½”, 12 ¼” and 8 ½”.

2. Drill Bits

The drill bit is one of the drilling equipment which has undergone the most changes above all equipment found in the drilling rig. It is the most important tool in the entire drilling phase, translating the surface horsepower into a brute force to crush or shear rocks. The drill bit has evolved throughout the decades and currently in the oil and gas industry, there are three main categories of drilling bits [1]:

- Roller cone bits
- Fixed cutter bits
- Hybrid bits

2.1. Roller Cone Bits

Roller cone bits have three major parts: cones, bearing and the bit body. Majority of them has three equally-sized cones which rotate independently as bit turns downhole. Generally, roller cone bits are used to drill a wide variety of formations, from very soft to very hard. Usually, the hard (high-compressive strength) formations are drilled using a short, closely spaced cone that chip and fracture the rock. The soft(low–compressive strength) formations are drilled using sharp, long teeth to gouge and scrap the rock [2]. Moreover, this type of bits can be classified as [3]:

- **Milled Tooth Bits** – have steel tooth cones, manufactured as an integrated part of a roller cone; teeth have carbide composite edges for wear protection; teeth size and shape depends on the formation type and hardness, the harder formation, the shorter and closely spaced teeth.
- **Tungsten Carbide Insert (TCI)** – have tungsten carbide teeth manufactured separately and squeezed into holes on the face of each cone, the harder formation, more rounded inserts.

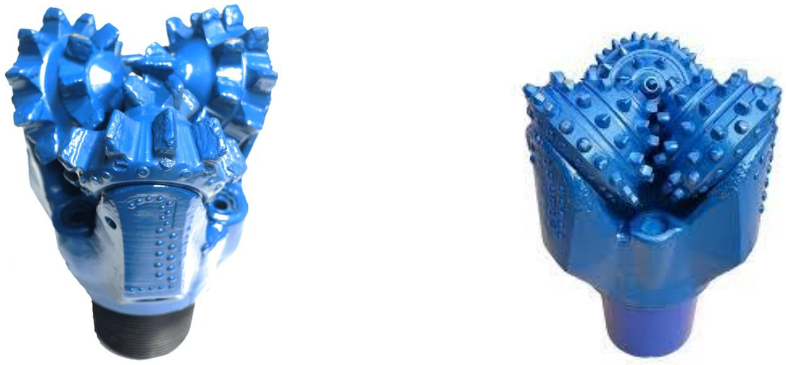


Figure 2.1 Rolling Cone Bits - Milled Tooth (left) and TCI (right) [4].

2.1.1. Bit Design

In general, the proper interaction between bit and formation is achieved by adjusting journal angle, cone shape, and cone offset. These elements control the cones rotations. Journals are axle-like items around which each cone makes a turn. The journal angle is an angle formed by the axis of the journal to a horizontal plane. The higher journal angle, the smaller the size of the cone. Also, the journal angle depends on the rock formation [5]:

- 33° - soft formations
- 34° - 36° - medium formations
- 39° - hard formations

Offset values, also known as skew angle indicates how much each journal is shifted to prevent the cone axis intersection in the middle of the bit. The bit with no offset value has an intersection point at the center of the bit. The offset value depends on rock formation type and usually is in the range from 0° in hard formation to 4° in soft formations [6].

Another important part of the roller cone bits are the bearings. The bearings allow relative motion between pin and cone. They are place on the pin, allowing cones to rotate during rock crushing. Bearings increase the operational reliability and overall effectiveness of the roller cone bit. There are three main types of bearings [6]:

- Sealed journal bearings
- Sealed roller bearings
- Sealed journal bearings

The last important part of the roller cone bits are the fluid nozzles. They improve hole cleaning as well as increase ROP by jetting mud at the bottom of the well to remove cuttings. The number and location of nozzles have an impact on bit performance, especially the relationship between ROP, bit cleaning and cutting removal. The ROP may be significantly increased by keeping nozzles angled to point drilling fluid straight to cones.

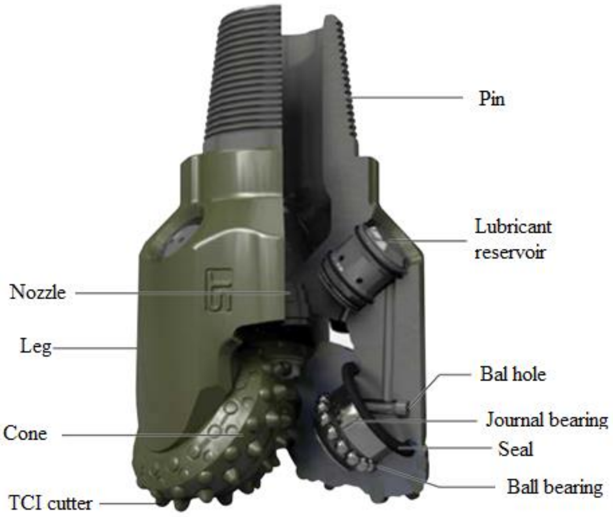


Figure 2.2 Major components of the Roller Cone Bit [7].

2.1.2. IADC Roller Cone Bit Classification

IADC developed the classification code which contains the three numbers and letter. The first three digits classify the bit in according to rock strength [8]. The code helps drilling engineers to describe what kind of drill bit they are looking for to the supplier.

- **First digit** – describes the bit type and formation hardness, Milled Tooth Bits have numbers 1 -3 (soft to hard formations) and Tungsten Carbide Insert Bits have number 4 – 8 (soft to hard formation)
- **Second digit** – describes the further breakdown of formation, numbers 1 – 4 (soft to hard formation)
- **Third digit –** describes the bit in according to bearing or seal type, numbers 1 - 7
- **Fourth digit –** describes additional bit features, for more complex tools more than one letter can be used

2.1.3. IADC Roller Cone Bit Dull Grading System

IADC also developed the system for classification of the bit dullness. After each run, the bit is meticulously inspected and evaluated. The proper evaluation of dull bit is critical for improving bit type selection and identifying those drilling parameters which can be modified to improve drilling performance and prolong the bit life. Every abnormal wear is recorded and measured to avoid excessive wear in the future. The system is intended to bring consistency across the drilling industry and to standardize the evaluation of certain bit characteristics. The bit dull classification consists of eight columns [9]:

- **Inner** – uses a number (0 – 8) to report the condition of cutting element which does not touch the wall of hole; describes the change from inner 2/3 of cutting structure,
- **Outer** – uses a number (0 – 8) to report the condition of cutting element which touches the wall of hole; reflects the importance of gauge and heel condition; describes the change from outer 1/3 of the cutting structure
- **Dull Characteristic** – uses two-letter code to report major dull characteristic of the cutting structure
- **Location** – uses a letter or number to report the location on the bit face where dull characterization occurs
- **Bearings** – uses a letter or number to report the bearing condition
- **Gauge** – reports the gauge of the bit or its reduction in 1/16th of an inch
- **Other Dull Characteristic** – reports any dull characteristic, uses the same two-letter as Dull Characteristic above
- **Reason Pulled** – reports the reason for bit run termination

2.2. Fixed Cutter Bits

Fixed cone bits rotate as one piece. Bit bodies are integrated with blades and cutters. They do not have any moving parts or bearings. Cones may be made from natural, synthetic or polycrystalline diamonds. They can be used to drill a wide variety of formations, from soft to very hard. Fixed Cutter bits remove formations through shearing motion. Moreover, this type of bits can be classified as [3]:

- **Polycrystalline Diamond Cutters (PDC)** – have small, round cones made from synthetic diamonds which can be easily attached to bit bodies, ensure better control in directional drilling than roller cone bits

- **Diamond Cutters** – have impregnated natural diamonds or TSP elements; as diamonds wear down, new diamonds are exposed to carry on the performance; fine-grained diamonds and coarse-grained are used to drill hard and very hard formations respectively

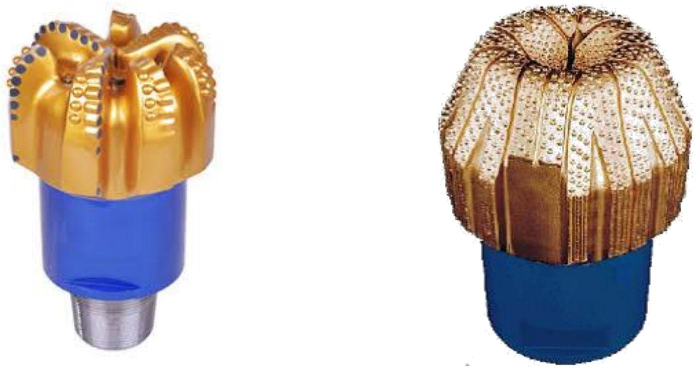


Figure 2.3 Fixed Cutter Bits – PDC (left) and Diamond (right) [10].

2.2.1. PDC Bit Design

Bit bodies are made from steel or matrix (tungsten). The selection of the body type depends on the operator's particular requirement. The ductility of the steel allows producing bit bodies with taller blades and large junk slots, which directs cuttings from the bit. Steel bodies bits are generally used for drilling in shales and soft formations. However, the steel bit bodies are less resistant to abrasion than the matrix body [7].

One of the most important characteristics of PDC bits is its profile shape. It shows the bit shape from the centre to gauge. Bit profile affects stability, durability, cleaning efficiency and ROP. Generally, the shorter profile the less stable and more aggressive bit is. Also, the larger nose radius, the more cuttings are produced at the nose, making a bit more aggressive. However, the durability and stability increase with profile and shoulder length.

Cutters in PDC bits are made from synthetic diamonds. The part of the cutter which interacts with rock formation is called the diamond table and is made from the carbide substrate. Diamonds cutters are exceptionally hard, have high wear–resistance and shear the rock formation easily. The bigger size of the PDC cutter, the more aggressive bit is as well as it reduces the cutter number and overall bit durability. Cutters orientation also has a big impact on the bit performance. The cutters orientation is described by back–rake angle. The smaller angle, the more aggressive bit it and can be used it softer formations as well. Back-rake high values increase wear resistance but decrease drilling efficiency [11].

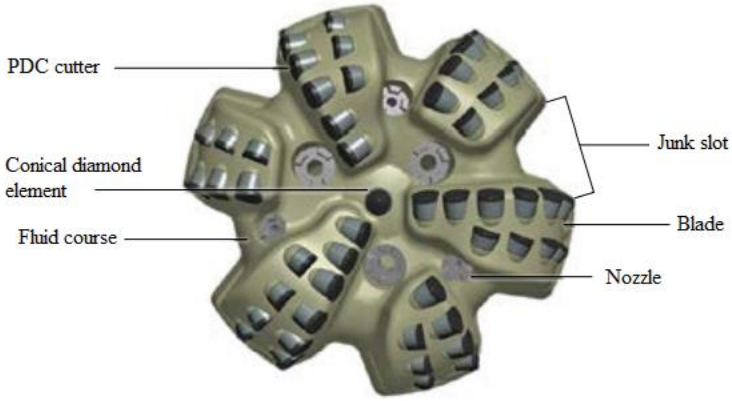


Figure 2.4 PDC bit face [7].

2.2.2. IADC PDC Bit Classification

Similarly to the roller cone bits classification, the IADC developed the classification code for PDC, TSP and diamond bits [12]. The code consists of one letter and three numbers. It allows the efficient bits selection for a particular rock formation.

- **First digit** – the letter describes the body part
- **Second digit** – describes the rock hardness to be drilled, number 1 – 8 (soft to hard formation)
- **Third digit –** describes the dominant PDC cutter size, number 1 – 4 (biggest to smallest sizes)
- **Fourth digit –** describes the bit profile, number 1 – 4 (shortest to longest profile)

2.2.3. IADC PDC Bit Dull Grading System

The IADC PDC bit dull grading system has similar principles as the roller cone bit dull grading system [13] shown in subchapter 2.1.3. The only difference is in the bearing/seals column. Due to the fact, that PDC does not have any bearings, the letter X is always put in this column. The detailed explanation of the nomenclature used in the Bit Dull Grading charts may be find in the *First Revision to the IADC Fixed Cutter Dull Grading System* [13].

Cutting structure				B	G	Remarks	
Inner Rows	Outer Rows	Dull Char.	Location	Bearings/Seals	Gauge 1/16th	Other Char.	Reason Pulled

Table 2.1 Bit Dull Grading Chart.

The Bit Dull Grading Chart is filled after every single bit run. The Appendix 1 shows the filled chart after one particular bit run as well as it shows the collected data after all runs in the well. The charts in Appendix 1 come from well F-7. Based on the chart there is possible to evaluate the bit performance in the run and compare it with previous runs or other wells.

In further calculations only the Inner Rows value is used as it is described in the models in the next subchapter. However, it would be interesting to evaluate the bitt dull grading based on the reason of pulling out, but the possessed dataset contains only several cases in which the Section Total Depth haven't been reached. Such small data variety is not sufficient for further Machine Learning approach.

2.3. Bit Wear Prediction Models

There are not many techniques to predict and evaluate the bit wear. The most common method is the aforementioned IADC code. It is an industry standard, however only describes the bit state before running it into the hole and after pulling it out. Moreover, it is highly subjective and the procedure depends on the engineer's accuracy and experience.

After a thorough investigation of the literature and SPE papers, more techniques were found to describe the bit state. The first discussed method was developed by B.Rashidi, G.Hareland, and R.Nygaard in 2008 [14]. The technique is based on the Borgouyne and Young Rate of Penetration (ROP) model, Mechanical Specific Energy (MSE) and rock drillability and is used to predict the real-time bit wear.

$$ROP = f_1 * f_2 * f_3 * f_4 * f_5 * f_6 * f_7 * f_8 \tag{1}$$

The $f_1 - f_8$ coefficients express the impact of different parameters on ROP such as rock drillability, bit wear and drilling parameters. The MSE describes how much energy is required to remove a given volume of rock and is further explained in the Chapter 6. The model also uses two constants K_1 and K_2 which are calculated based on the offset data and the input to the formulas below.

$$h = \frac{(Depth_{current} - Depth_{in})}{(Depth_{out} - Depth_{in})} * \frac{DG}{8} \tag{2}$$

$$Norm\left(\frac{1}{K_1}\right) = 1 - h^b \tag{3}$$

$$B = 5{,}6392 * h + 0{,}4212 \tag{4}$$

where:

$DG - reported\ bit\ wear\ dullness$

$h - fractional\ bit\ teeth\ dullness$

The model calculates the bit wear based on the K_1 constant. In order to correlate the K_1 trends with bit wear grade the normalized inverted K_1 is adjusted against bit wear. Then, by using regression methods the best B constant is found and equation (4) is inserted into equation (3) to estimate the bit wear for real–time situations.

However, the model has few shortcomings. The first one is the use of the ROP model which is based on different constants and as many regression models it has limited prediction capability. The second one is the K_1 and K_2 constants which are quite difficult to determine and the obtained results may significantly vary between the surveys.

The second method was developed by Z. Liu, C. Marland, D. Li and R.Samuel in 2014. It is an analytical method which is based on parameters like ROP, Weight on Bit (WOB), RPM and the confined compressive rock strength. The technique takes into account the inverse pyramid approximation of the PDC bit cutter and Gamma Ray log in order to investigate the formation influence on the bit wear. Not getting much into the details in the derivation of equations some of the final formulas are presented below.

$$\frac{\Delta h}{h} = \sqrt[3]{\frac{\pi * \beta * D_b^2 * \alpha * S^2 * X}{3{,}2 * V_o * G * (1 - \left(\frac{\Delta h}{h}\right)_{i-1})} + \left(\frac{\Delta h}{h}\right)_{i-1}^{3}} \tag{5}$$

$$W_f = 1 - \frac{\Delta h}{h} \tag{6}$$

$$\Delta BG = 8 * \frac{\Delta h}{h} \tag{7}$$

where:

$\beta - abrassive\ cnstant\ [-]$

$D_b\ - bit\ diameter\ [in]$

$\alpha - normalized\ rock\ content\ [-]$

$S - confined\ compressive\ strength [psi]$

$X - depth\ increament\ [ft]$

$V_o - volume\ of\ truncated\ cylinder\ with\ flat\ surface\ [-]$

$\frac{\Delta h}{h} - fractional\ bit\ wear\ [-]$

$G - model\ constant\ [-]$

$W_f - bit\ wear\ function\ [-]$

$\Delta BG - bit\ wear\ grade\ [-]$

The presented model looks very promising and according to their authors, the IADC bit dull grade is properly calculated by the model. However, due to lack of some geological parameters, it was impossible to calculate or predict the confined compressive strength of the formations. Therefore, the model has not been tested in this thesis.

The last method for bit wear prediction is presented in the *Applied Drilling Engineering* textbook [2]. Unfortunately, the method works only for the roller cone bits, however, the equations were modified in one of the master thesis [16] to be able to predict the PDC bit as well.

The model uses parameters like WOB, RPM and Drilling Time for bit wear calculation and also consists of some constants related to the type of bit which was used in the well. The equations below show the instantaneous rate of tooth wear for roller cone and PDC bits respectively.

$$\frac{\Delta h}{\Delta t} = \frac{1}{\tau_H}\left(\frac{N}{60}\right)^{H_1}\left[\frac{\left(\frac{W}{d_b}\right)_m - 4}{\left(\frac{W}{d_b}\right)_m - \left(\frac{W}{d_b}\right)}\right] * \left(\frac{1+\frac{H_2}{2}}{1+H_2 h}\right) \tag{8}$$

$$\frac{\Delta h}{\Delta t} = \frac{H_3}{\tau_H}\left(\frac{N}{160}\right)^{H_1}\left[\frac{\left(\frac{W}{d_b}\right)_c}{\left(\frac{W}{d_b}\right)_m}\right] * \left(\frac{1+\frac{H_2}{2}}{1+H_2 h}\right) \tag{9}$$

where:

$h - fractional\ tooth\ height$

$t - time\ [hrs]$

$H_1, H_2, H_3, \left(\frac{W}{d_b}\right)_m - constants\ [-]$

$W - bit\ weight\ [1000\ lb_f]$

$N - rotary\ speed\ [rpm]$

$\tau_H - formation\ abrasivennes\ constant\ [-]$

The parameter J_2 is introduced in order to estimate the formation abrasiveness constant. The equations below are for roller cone and PDC bits respectively.

$$J_2 = \left[\frac{\left(\frac{W}{d_b}\right)_m - \left(\frac{W}{d_b}\right)}{\left(\frac{W}{d_b}\right)_m - 4}\right] * \left(\frac{60}{N}\right)^{H_1} * \left(\frac{1}{1+\frac{H_2}{2}}\right) \tag{10}$$

$$J_2 = \frac{1}{H_3}\left[\frac{\left(\frac{W}{d_b}\right)_m}{\left(\frac{W}{d_b}\right)_c}\right] * \left(\frac{160}{N}\right)^{H_1} * \left(\frac{1}{1+\frac{H_2}{2}}\right) \tag{11}$$

Each of the equations above can be expressed by:

$$\int_0^{t_b} dt = \tau_H * J_2 * \int_{h_i}^{h_f} (1 + H_2 h)dh \tag{12}$$

Integration of this equation gives the result.

$$t_b = \tau_H * J_2 * (h_f - h_i + \frac{H_2}{2} * (h_f^2 - h_i^2)) \tag{13}$$

where:

$t_b - bit\ hours\ on\ bottom\ [hrs]$

$h_i - initial\ tooth\ wear\ ratio\ [-]$

$h_f - final\ tooth\ wear\ ratio\ [-]$

The initial and final tooth wear ratio are taken from the IADC Bit Dull Grading Chart. However, one of the model requirements is to convert the IADC number and divide them by 4. Therefore, instead of using the IADC scale $0-8$, the model uses the range between $0-2$. Having known the initial and final bit wear it is possible to calculate the bit wear at each step. Solving for the abrasiveness constant

$$\tau_H = \frac{t_b}{J_2 * (h_f - h_i + \frac{H_2}{2} * (h_f^2 - h_i^2))} \tag{14}$$

Due to the fact, that the drilling parameter can vary during the drilling phase, the abrasiveness constant is calculated as the sum over time intervals Δt_b.

$$\tau_H = \sum \frac{t_b}{J_2 * (h_f - h_i + \frac{H_2}{2} * (h_f^2 - h_i^2))} \tag{15}$$

Finally, assuming that H_2 coefficient equals 1, the bit wear at any time is calculated based on the formula:

$$h_j = \sqrt{1 + \frac{2*t_{b_j}}{\tau_H + J_{2_j}} + 2 * h_{j-1} + h_{j-1}^2} - 1 \tag{16}$$

Having all the necessary parameters for the equations, the model was chosen for the application in the bit dull grading prediction. It contains separate formulas for roller cone and PDC bits, which may be used in different well sections with better results.

3. Geology

3.1. Formation Evaluation

Currently, the drilling optimization is one of the key topics in the oil and gas industry. One of the many parts in this process is the ability to classify the drilled formations based on the drilling data in order to reduce the drilling time and drilling problems.

The formation classification would enable to optimize the real-time operations. Knowing the formation, it will be possible to estimate the pore pressure as well as the ROP could be optimize in order to drill as fast as possible or to prevent the hole instability problems. Moreover, it will be extremely beneficial in the geosteering and enable to stay within the reservoir increasing the contact between the well and the reservoir. This will allow to increase the hydrocarbon flow in the production phase of the well cycle.

3.2.The Volve Dataset

3.2.1. Disclosed Data

Equinor disclosed all subsurface and production data in June 2018. This dataset consists of around 40 000 files covering every single phase of the field [17]. The most important folders cover well data, real-time drilling data, daily reports as well as logs and final well reports. This comprehensive and complex dataset is a perfect test ground for further formation classification and bit dull grading prediction case study.

3.2.2. General Information

The Volve field is a relatively small oil discovery. It is approximately 2 x 3 km four-way closure situated on structural high within Sleipner area in the North Sea. The water depth is in range between 85 to 95 meters. The Volve field has many geological similarities with the neighboring structures Loke and Sleipner Øst. The field was discovered in 1993 and the appraisal procedure took place in 1996 and 1997 [18]. The plan for development and operation (PDO) was designed and approved in 2005. The entire field was expected to produce only for 3 – 5 year, however it was shut down in 2016 exceeding the initial expected production live. The decommissioning phase started in 2018.

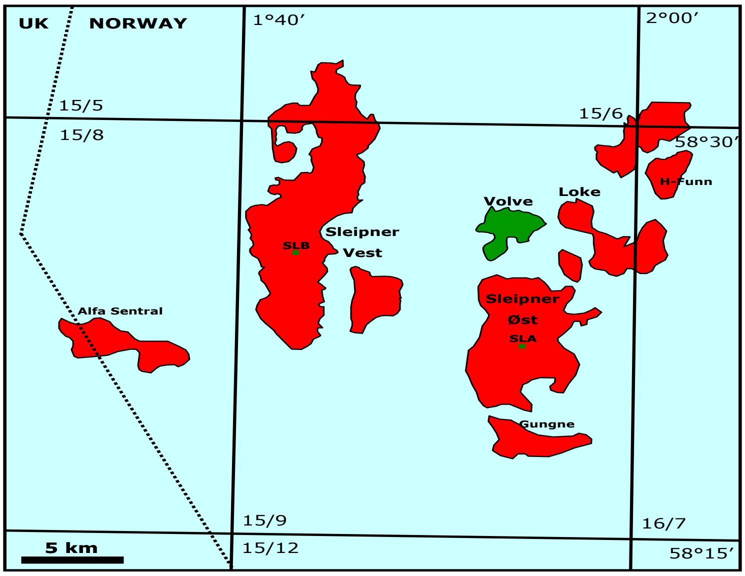


Figure 3.1 The location of Volve Field [16].

3.2.3. Geology

The Volve field produced oil from Jurassic sandstone of Hugin Formation. The reservoir was located at depth of 2750–3120 meters. There are large lateral thickness variations in Hugin Formation which are mainly caused by laterally varying subsidence during deposition. The evolution of the Volve structure was largely controlled by salt tectonics, affecting the Hugin reservoir deposition. The oil in the Volve field has been sourced from uppermost organic-rich claystone of the Draupne formation. The kitchen area is the Sleipner graben located only 5-10 km west and northwest from Volve [19].

In terms of the drilling-related and rock mechanics issues, the Hordaland shales are normally associated with a high smectite content which may lead to instability and higher pore pressure. However, the Grid sand is also present, preventing pressure from reaching very high values. Due to high shale content in wellbore, especially in high angle parts may cause severe instability.

The Balder formation contains loose friable tuff which may cause mud losses and is prone to washouts. Tuff may also act as the unstable formation and possess relatively low fracture gradient. The Cromer Knoll group lead to many challenges with respect to tight hole and collapse, especially in the Sola formation. The general lithology for singular well is presented in the figure 3.3 [20].

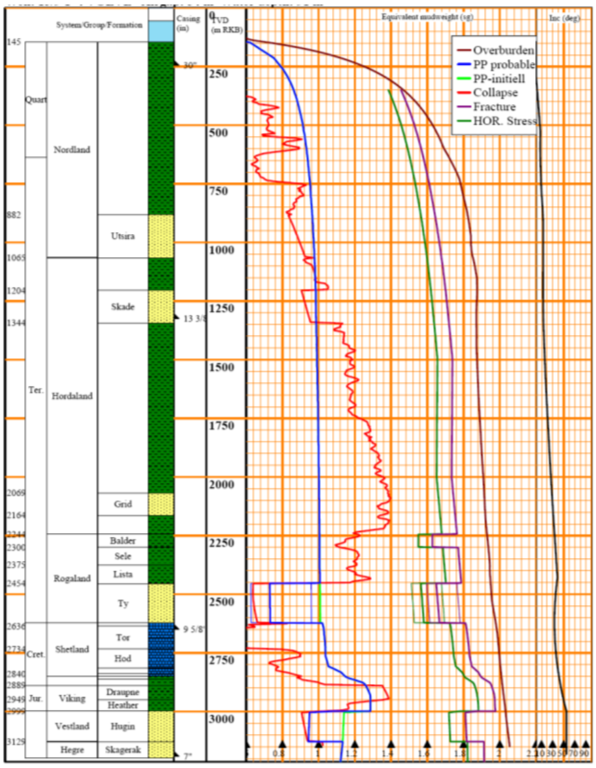


Figure 3.2 Well F-4 – pore pressure and stability prognosis [20].

3.2.4. The Wells

The wildcats were drilled in late nineties, when the measuring equipment and data processing capabilities were not as good as today. Hence, the work has been focused on the wells drilled in the XXI century. The 9 wells in total – F-1, F-4, F-5, F-7, F-9, F-10, F-11, F-14 and F- 15 were used for analysis. Depending on the well purpose – production, injection, observation – they have sections: 36", 26", 17 ½", 12 ¼" and 8 ½".

The wells were designed to maximize the production from the Hugin formation. During the drilling phase, the geosteering were used in order to maximize the reservoir length and connect the different fault block. Generally, the trajectory of the injection and observation wells are usually close to J-shape, while majority of the production wells are the multilaterals. The example of the injection well is shown in the figure 3.4 and the production wells is shown in the figure 3.5 and 3.6. Majority of the wells have the Total Vertical Depth (TVD) around 3100 – 3400 meters, but the Measured Depth (MD) varies a lot. Usually, the longer lateral section, the higher MD. The dogleg hasn't been higher than $6°/30\ meters$.

Lithology	**Description**
Seabed	Consists of dense to very dense sands overlaying stiff clay.
QUARTERNARY	Clay with thin stringers of sand. Coarser material up to boulder size may occur.
NORDLAND GP. - *Pliocene and Pleistocene*	Grey claystone with thin stringers of sand and siltstone.
Utsira formation	Fine to medium-grained, moderately well and well-sorted sandstone with minor silt and limestone stringers.
HORDALAND GP. - *Eocene to Miocene*	Dominated by claystone and minor limestone/dolomite stringers with exception of the sandy Skade and Grid formations.
Skade Fm.	Medium-grained and moderately sorted sandstone, occasionally calcareous cemented.
Grid Fm.	Very fine to fine-grained sandstone.
ROGALAND GP. - *L. Paleocene to L. Eocene*	
Balder Fm.	Vari-colored claystone, partly tuffaceous with some limestone stringers.
Sele Fm.	Claystone and minor limestone stringers.
Lista Fm.	Noncalcareous claystone with minor limestone stringers.
Ty Fm.	Very fine to medium-grained sandstone, moderately to poor sorted, with some interbedded claystone, siltstone, and a few limestone stringers.
SHETLAND GP. – *U. Cretaceous*	
Ekofisk Fm.	Chalky off-white to light grey limestone, moderately hard with traces of claystone and sandstone.
Tor Fm.	White limestone, moderately hard becoming pale red-brown and very hard with depth, traces of claystone.
Hod Fm.	Off-white to white limestone, moderately hard, chalky, grading to marl with depth, glauconite.
Blodøks Fm.	Medium to dark grey marl, argillaceous laminations, glauconitic in parts.
Hidra Fm.	Off-white firm limestone.
CROMER KNOLL GP. - *U. Cretaceous to L. Cretaceous*	
Rødby Fm.	Marl with argillaceous laminations.
Sola Fm.	Marl and claystone.
Åsgard Fm.	Interbedded limestone and marl with some minor layers of claystone and siltstone.
VIKING GP. – *U. Jurassic*	
Draupne Fm.	Very organic-rich claystone, micaceous, carbonaceous and traces of pyrite.
Heather Fm.	Claystone with limestone stringers and interbedded claystone, kaolin, sandstone, and limestone in the lowermost part.
VESTLAND GP. – *M. Jurassic*	
Hugin Fm.	Sandstone, very fine to very coarse-grained, moderately to well sorted. Rare claystone stringers.
Sleipner Fm.	Sandstone, very fine to medium grained, moderately to well sorted, grey claystone and layers of coal.
HEGRE GP. – *U. Triassic*	
Skagerrak Fm.	Fine-grained sandstone with some interbedded silty sections.
Smith Bank Fm.	Reddish brown claystone with occasionally sandstone stringers.

Figure 3.3 Well F-4 general lithology [20].

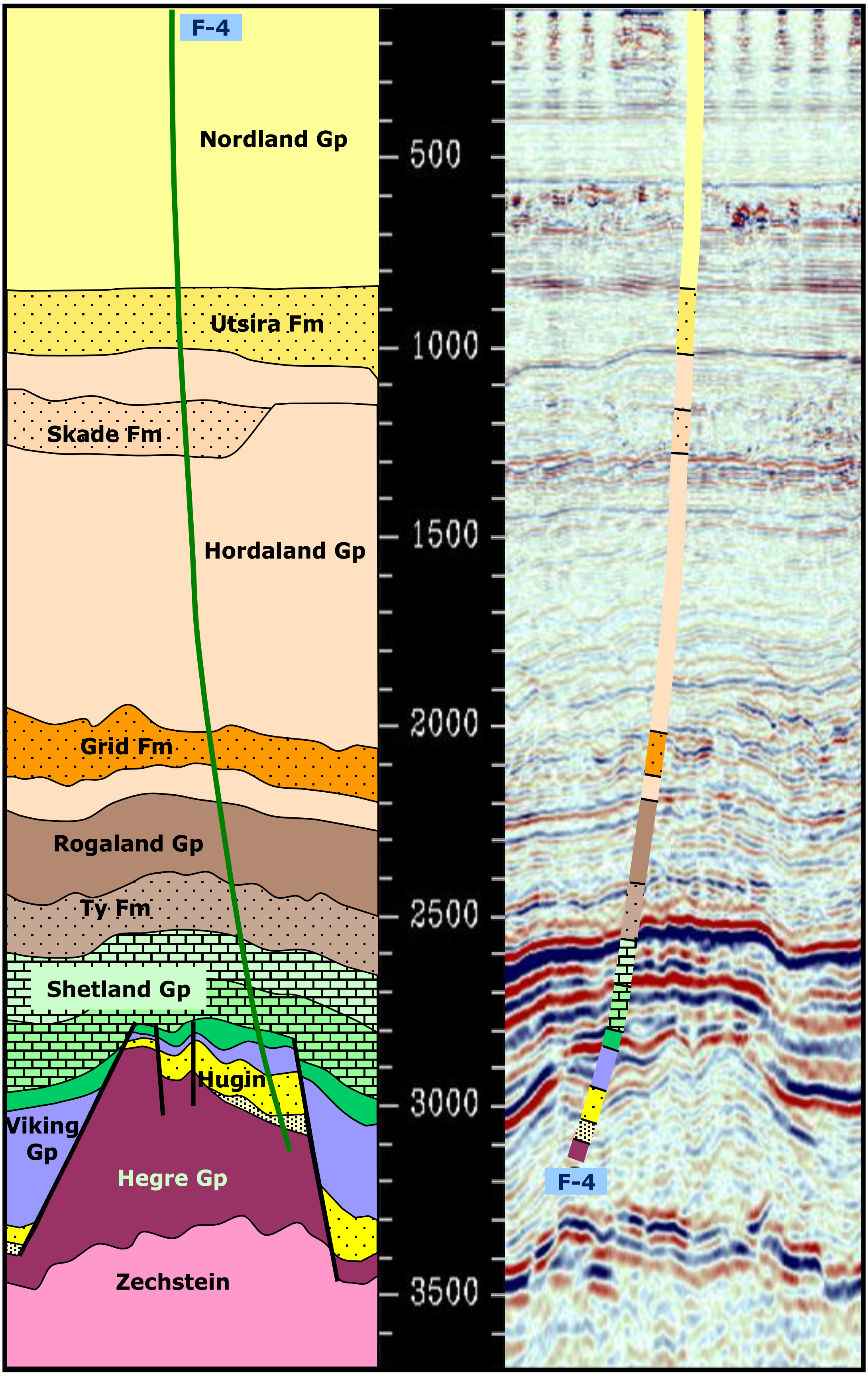


Figure 3.4 Well F-4 geological and seismic cross section [20].

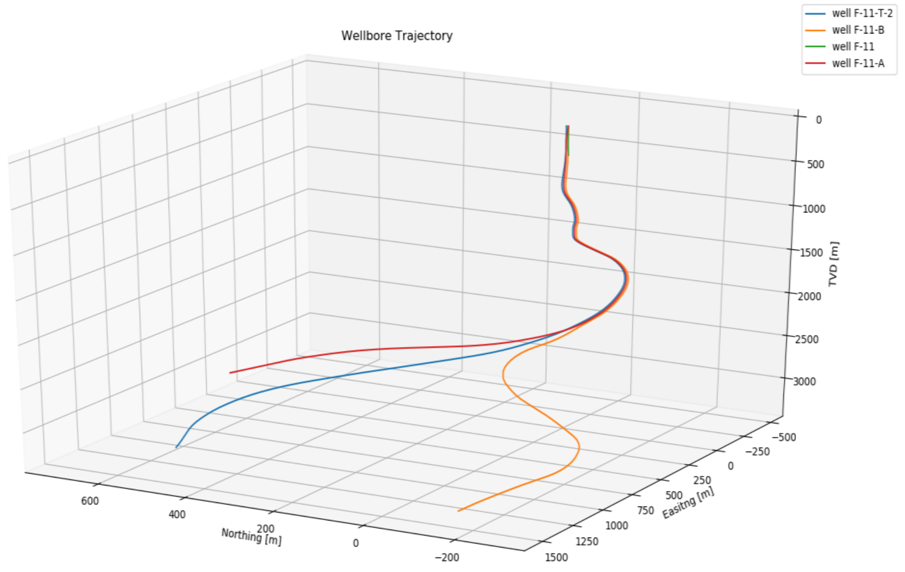


Figure 3.5 The Well F-11 trajectory.

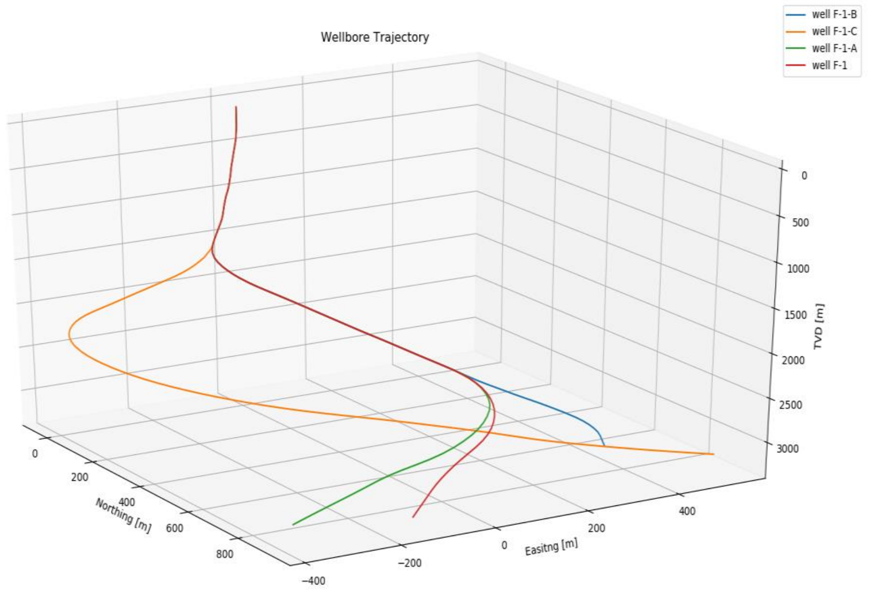


Figure 3.6 The Well F-1 trajectory.

3.2.5. Drilling Problems

Generally, the entire field were drilled without any major problems. There were only several bit runs which haven't reached the section total depth. Mostly, the reason of pulling out the hole (POOH) wasn't connected with drilling related problems such as low ROP or bit worn-out, but it was pulled due to malfunctions with MWD or gathering the data. One of the POOH reports is shown in the Appendix 2. The figure gives the brief explanation why the bit was worn-out so early and shows the recommendations for solving such a problem in the future. The bit dull grading indicates that bit was worn-out quite severe (inner rows – 4, outer rows – 3) and characterize the bit state in accordance to table 2.1 and *First Revision to the IADC Fixed Cutter Dull Grading System* [13].

Having analysed the well reports it may be concluded that drilling parameters were chosen properly. However, such low number of POOH can be also caused by limiting the ROP due to cuttings handling problems. Lower ROP may have diminished the drilling problems related to formation issues such as pack-off, stick-slip or excessive bit dullness. Also, it could have positive impact on bit life prolongation.

4. Data Analytics

The main objective of this thesis is to create the machine learning models for formation classification and bit dull grading. In order to fulfil the goals, the data-driven approach has been used. Such an approach uses scientific methods and algorithms to extract data and make decisions based on data analysis and interpretation.

The data analytics is the process of analysing the raw data in order to make conclusions about the information they contain. Majority of the processes are carried out using specialized algorithms and software. These techniques can reveal patterns and trends, which otherwise would be omitted in the immense flow of information. Then, the possessed information about the trends and patters may be used to increase the system productivity and business performance. The data analytics process can be divided into several steps:

1. Determine how the data is grouped.
2. Collect and process the data.
3. Organize the collected data and clean up before analysis.
4. Develop and evaluate the model.
5. Deployment.

One of the key things in the data analytics process is to correctly define the problem as well as its overall sound understanding. This allows to select appropriate parameters form the available data and in case when some features are missing in the dataset, to calculate similar parameters that will significantly increase the quality of the subsequently created models.

4.1. Choosing the Right Environment

Having known the data analytics tools as well as complexity of the Volve dataset, the Python programming language was chosen to create models and run calculations. It is said that Python is the best coding language for data mining and analysis. Additionally, it has a huge community, so if any obstacle is encountered it may be easily overcome thanks to the information posted on specialized forums. This environment contains many powerful libraries, ranging from basic statistics to complex machine learning algorithms [21]. All libraries excel in performance, productivity and the ability to collaborate, making the whole workflow of data handling and visualization quite straight-forward compared to other languages.

4.2. Data Preparation and Selection

Due to the immense size of the dataset, the first challenge was to get familiar with the available data. In order to be able to read the relevant data, the dedicated XML files were created to be able to automatize the process of reading the daily drilling reports as well as well logs. In the Real-Time Drilling Data folder, the most valuable files were Drilling Depth well logs which consist of the basic drilling parameters such as Rate of Penetration (ROP), Weight on Bit (WOB), Torque, RPM, Flow Rate, MD, TVD. The code with the logs extraction is attached in Appendix 3.

Having basic knowledge of the dataset, the depth based data were chosen to create the dataset. The choice was made based on the available data and the understanding of the data. Unfortunately, the time based does not have clearly explained the rig activity, so finding only the drilling phase would be a challenge. The final well reports were read to have a bigger picture of the situation in the wells – F-1, F-4, F-5, F-7, F-9, F-10, F-11, F-14 and F-15. Those reports contain valuable data about lithology, mud and drilling parameters. The reports and well history were digitalised by one of the University of Stavanger student and the detailed information about the wells can be found by using the link in the reference [22]. In order to have proper datasets, the dedicated MS Excel spreadsheets were created for each well respectively The spreadsheet is divided into sheets based on the bit runs to be able to predict the bit dull grading. The sample print screen of MS Excel files is attached in Appendix 4.

Unfortunately, not all data was stored in the XML files, but some of them were only stored as PDFs. The XML files were automatically read and saved as MS Excel files format, while data in PDF format were manually rewritten to same spreadsheets. After having all the necessary parameters, the spreadsheets were loaded to the written code. The best library in Python to handle data is Python Data Analysis (pandas). Pandas is an open source, easy-to-use tool which conducts all necessary operations on datasets. It increases productivity and enhances the performance of the whole code without writing complex algorithms.

Having analysed the available data in the dataset and knowing the drilling phase physics, some new parameters were calculated. This step would create additional input data for both the formation classification and the bit dull grading prediction part. This will not only describe more realistically the condition in the well during the drilling phase, but also it will extend the number of the robust parameters which give the better Machine Learning models performance. Due to the lack of the literature, the extra parameters where needed for the bit dull grading prediction part. Therefore, a couple of meetings were held with the drill bit engineers to find out what parameters have the greatest impact on drill bit wear.

The first parameter is the Mechanical Specific Energy (MSE) which tells how much work is done to excavate a volume unit of rock. The equation was introduced by Taele in 1965 [23]. Taele's formula is an appropriate parameter for formation classification. The harder the formation, the more resistance is, hence the MSE value should be higher.

$$MSE = \frac{WOB}{A_B} + \frac{120 * \pi * RPM * TQ}{A_B * ROP} \; [psi] \tag{17}$$

where:

$WOB - weight\ on\ bit\ [lbs]$

$A_B - bit\ area\ [in^2]$

$RPM - revolutions\ per\ minute\ [-]$

$TQ - torque\ [lb_f]$

$ROP - rate\ of\ penetration\ [ft/hr]$

The next parameter is the Depth of Cut (DoC) [24]. The parameter describes how deeply the drill bit cuts per revolution. Generally, the DoC values below $1[mm/rev]$ indicated the instability problems such as bit whirl.

$$DoC = \frac{ROP * k}{RPM} \; [mm/rev] \tag{18}$$

where:

$ROP - rate\ of\ penetration\ [m/hr]$

$RPM - revolutions\ per\ minute\ [-]$

$k - conversion\ factor = 16{,}66\ for\ metric\ units$

The another parameter is Bit Aggressiveness (BA) [25]. The parameter is determined by the cutters exposure and angle. The more aggressive the bit, the more prone is to change direction while drilling.

$$BA = \frac{36 * TQ}{WOB * A_B} \; [-] \tag{19}$$

where:

$WOB - weight\ on\ bit\ [lbs]$

$A_B - bit\ area\ [in^2]$

$TQ - torque\ [lb_f]$

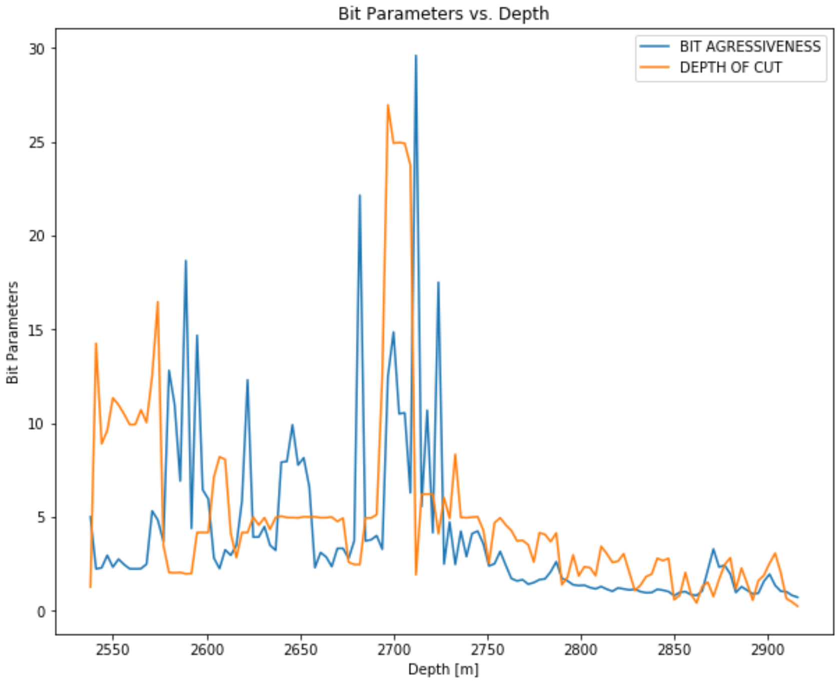


Chart 4.1 Bit Parameters versus Depth for
well 15-9-F-11-B 12 ¼" section.

Total Energy (TE) and Revolutions (REV) made by the drill bit in order to drill the specific depth interval are two commonly used parameters in the industry to evaluate the drill bit state.

$$TE = \frac{WOB * krev}{D_B} \; [-] \tag{20}$$

$$krev = \frac{RPM * depth\ drilled}{\frac{ROP}{60\ min}} \; [-] \tag{21}$$

where:

$WOB - Weight\ on\ Bit\ [klb]$

$RPM - revolutions\ per\ minute\ [-]$

$ROP - rate\ of\ penetration\ \left[\frac{ft}{hr}\right]$

$krev - revolutions\ used\ to\ drill\ unit\ of\ depth\ [ft]$

$D_B - bit\ diameter\ [in]$

These parameters, especially when used in cumulative form are able to give some information about the bit state. They do not give a precise answer about the bit wear, but based on them is possible to form an impression about the bit state and bit performance.

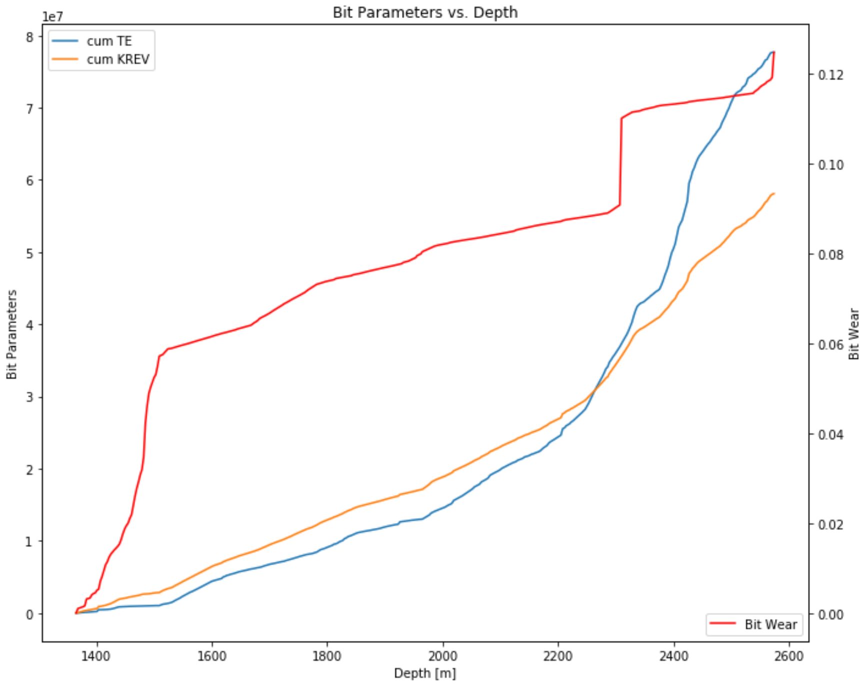


Chart 4.2 Cumulative TE and KREV versus Depth for well 15-9-F-11 17 ½ ” section.

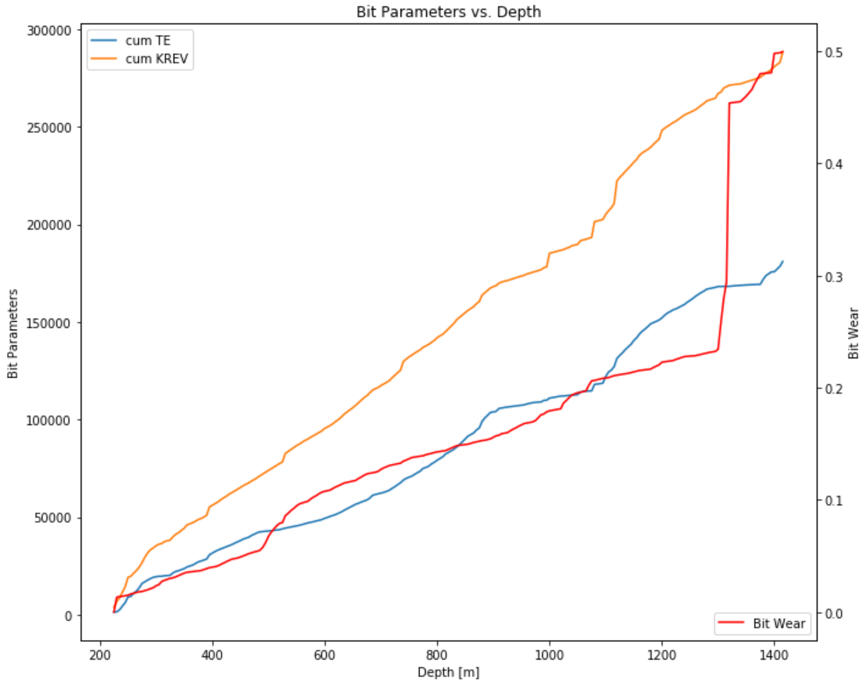


Chart 4.3 Cumulative TE and KREV versus Depth for well 15-9-F-5 17 ½ ” section.

The previous charts shows the cumulative TE and REV. The Chart 4.2 shows the bit run in which bit was chosen properly (two first IADC digits were 0 and 0) while the Chart 4.3 shows the bit run in which bit was quickly worn-out(two first IADC digits were 4 and 3). The difference in the parameters behaviour is clearly seen and both TE and REV grow quickly while the bit is not working properly. It may be useful to implement the TE and REV real-time monitoring in the drilling operations to choose the proper time of pooling bit out of the hole.

The next parameters are the Bit Nozzle Velocity, Impact of Jet Nozzles on Hole Bottom and Cross Flow Velocity under the bit [26]. It describes the fluid velocity which escapes from the bit nozzles. In softer formations usually encounter in the initial well sections, fluid may contribute to ROP and increase the drilling speed. Moreover, while using PDC bits flow rate has a critical impact on the cutters cooling and enhancing the bit life.

$$V_n = 0{,}321 * \frac{Q}{TFA} \left[\frac{ft}{s}\right] \tag{22}$$

$$JIF = \frac{MW * Q * V_n}{1930} \left[lb_f\right] \tag{23}$$

$$V_c = \sqrt{\frac{108{,}5 * Q * V_n}{N_N * D_B}} \left[\frac{ft}{s}\right] \tag{24}$$

where:

$V_n - nozzle\ velocity\ [\frac{ft}{s}]$

$JIF - impact\ of\ jet\ nozzles\ on\ hole\ bottom\ [lb_f]$

$V_c - cross\ flow\ velocity\ under\ the\ bit\ [\frac{ft}{s}]$

$Q - flow\ rate\ [gpm]$

$TFA - total\ flow\ rate\ [in^2]$

$MW - mud\ weight\ [ppg]$

$N_N - number\ of\ nozzles\ [-]$

$D_B - bit\ diameter\ [in]$

As may be seen most of the formulas are flow rate dependent. It only underlines how this parameter is important for the drilling process and must be treated with caution. The figure below shows the calculated parameters for the sample well.

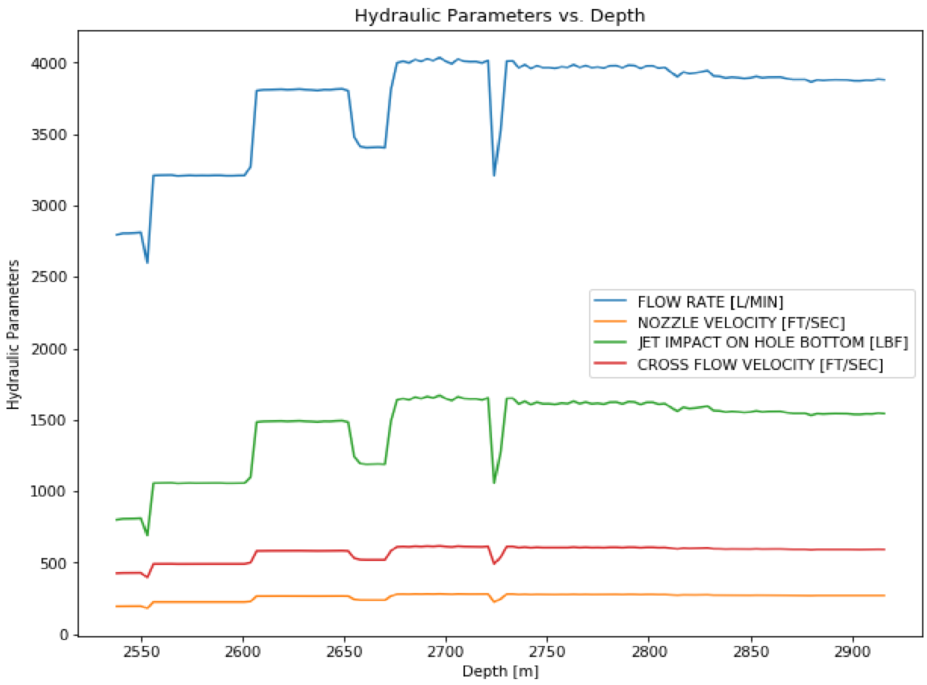


Chart 4.4 Hydraulic Parameters versus Depth for well 15-9-F-11-B 12 ¼” section.

4.3. Drilling Data Quality

Thanks to the measurement apparatus there is a possibility to obtain the parameters which describe the drilling process and its performance. The sensors are located both in the Measurement While Drilling(MWD) tool or at the surface and transmit the data to the main computer. However, the measurement apparatus located downhole usually need to cope with high temperature and pressure. Also, taking into account the longer wells and more sophisticated measurement tools, working in such an inhospitable environment may lead to problems with sensors, electronics, data gathering processor may cause sudden gaps in data transition. Hence, it is important to have a sound understanding of the dataset, parameters and its range. The ability to remove the extreme observation points is the key factor in data handling process. It may enable to prepare the robust dataset which increases the model’s performance.

The fastest method of identifying and removing the observation points which are distant from the rest of the data is based on mathematical methods. However, the computer would follow it blindly and some valuable data points may be lost. In order to avoid such a loss, the user should first plot the data and try to understand the current dataset. Then, having understood the data some automatic methods may be used or outliers in the most important parameters should be removed manually.

4.4. Outlier Removal

After creating the dataset, the next step is to clean it. The cleaning process ensures that the remaining data represents the problem in the best possible way. Datasets often contain points that are distant from other points and unlike the other data. These extreme observations are called outliers and can skew or mislead the training part of the machine learning process. The result is longer training time as well as a less accurate model gives poorer results. The model accuracy and performance may be easily improved by removing outliers, but the whole process must be conducted meticulously. Usually, the outliers may come from [27]:

- Measurement or input error
- Data corruption
- True outlier observation

In data science, there is a variety of methods to define and identify the outliers ranging from statistical approaches throughout distance-based approaches up to high-dimensional approaches. In this study, only four basic methods will be described.

4.4.1. Scatter Plot

The scatter plot is considered to be the simplest method to detect the outliers. It simply plots value for typically two variables from the dataset. The dataset is displayed as a collection of points, each having the one variable determine horizontal and vertical axis respectively. By looking at the plot is relatively easy to detect the outlier, however, the outlier removal after using this method is more complex and time-consuming.

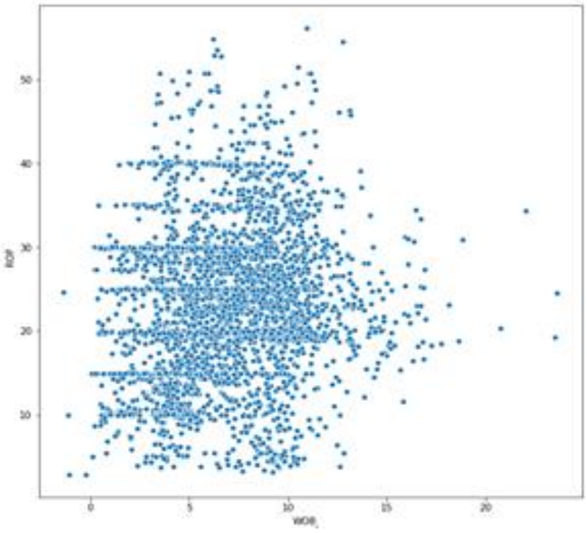


Chart 4.5 Scatter plot.

4.4.2. IQR Score

The interquartile range – IQR – is the statistical method widely used to identify outliers. The interquartile range is the range between the first and the third quartiles. It is considered that any data point that is located outside of either 1,5 times the IQR below the first or 1,5 times the IQR above the third quartile is outside the dataset and may be considered as the outlier [28]. In Python, the graphical representation of the IQR method is the boxplot from the seaborn library where outliers are shown as black dots. In order to better understand the dataset and see the representation of distribution is to plot boxplot with swarm plot on the same plot. The code in Appendix 5 shows the IQR method.

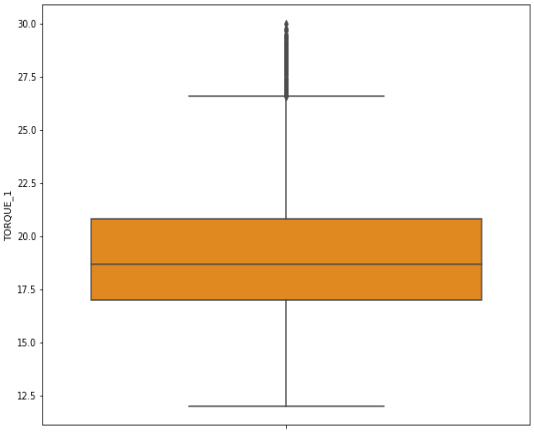


Chart 4.6 Boxplot.

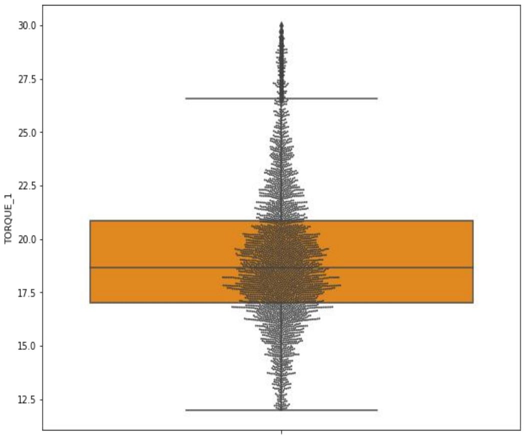


Chart 4.7 Boxplot with swarm plot.

4.4.3. Z Score

The Z-score is also known as a standard score describes the observation point in terms of its relationship with a mean and standard deviation of the datasets. The standard score is finding the distribution of data where mean is 0 and the standard deviation is 1. The scores range from -3 standard deviations – fall too far left, up to 3 standard deviations – fall to far right of the normal distribution curve. If the value of the standard score is greater than 3 or lower than -3, the observation point is considered to be an outlier [29]. The code in Appendix 6 shows how to identify the outlier by using the Z-score method. The output is the two array where first contains the list of row numbers while second contains the list column number where Z-score is higher than 3.

$$z_i = \frac{x_i - \bar{x}}{s} \tag{25}$$

where:

$z_i - z_{score}$

$x_i\ - observation\ point$

$\bar{x} - dataset\ mean$

$s - dataset\ standard\ deviation$

4.5. Feature Selection

Datasets attributes are hardly equal and have a different impact on the created model, hence the feature selection process is one of the core concepts of data handling. Thanks to that, it is possible to select these features in the dataset which contributes most to the output and avoid choking the model. The relevant features improve the model accuracy, performance and further, reduce the time needed to teach the model in machine learning techniques. Besides, there are more benefits of performing feature selection after creating the dataset and before running the model[30]:

- Reduced overlapping – less unwanted data decrease the chance to make decisions based on noise
- Improved accuracy –fewer attributes increase the model accuracy
- Reduced training times – based on a smaller dataset algorithm will be trained faster

Generally, there are three groups for selecting the features.

4.5.1. Filter Methods

Filter feature selection methods do not incorporate learning. They use the statistical tools to assign a score to each feature. After that, all features are ranked by the given score. These methods are often univariate and consider the dependent variable [30]. One of the best examples is the chi-squared test, univariate feature, and correlation coefficient scores.

4.5.2. Wrapper Methods

Wrapper feature selection methods use a learning process to measure the quality of features combinations. The process starts when various features sets are created, evaluated and finally compared with other sets. The combinations of features are evaluated by using the predictive model which assign a score based on its accuracy. Different methods may be used in the search process varying from methodical algorithms such as best-fit search, throughout stochastic like random hill-climbing algorithms up to heuristics methods such as forward and backward passes to handle features. The recursive feature elimination algorithm is a good example of a wrapper method. The wrapper methods are model oriented and usually gets a good performance for the chosen model. Unfortunately, they are computationally expensive in comparison with other methods [31].

4.5.3. Embedded Methods

In the embedded methods the learning part and feature selection part cannot be separated. The algorithm learns which features contribute most to the model accuracy. The learning process is being done while the model is created [31]. The regularization methods are one of the most common types of embedded methods. These methods are computationally demanding.

4.5.4. Choosing the Proper Techniques

Due to the fact that possessed datasets do not have immense size, two techniques were chosen from the aforementioned methods:

- **Extra Tree Classifier** – is a part of the scikit library. The method output is the feature name, score rank and feature score in percentages. The technique give easy to understand representation of the features and the user have the possibility to evaluate whether the already possessed feature are sufficient or whether more features should be added to the model
- **Correlation Matrix with Heatmap** – it shows how the features are related to each other or to the target variable. The correlation can be either positive (increase in feature increases the target variable) or negative (increase in feature decreases the target variable). The user can easily identify which of the features are most related to the target variable.

Both the formation classification and the bit dull grading prediction case used the techniques described above. Unquestionably, in the feature selection process, apart from understanding of the problem, the meetings with the industry representatives where extremely beneficial. Their detailed opinions helped to choose the proper artificial features and took a look how similar problems are solved in the industry. Then, the artificial features where calculated as described in the subchapter 4.2 and later used in the process of creating the machine learning models.

5. Machine Learning

Machine learning (ML) has evolved from computer science and enables to design the algorithms that are able to learn from experience and make decisions without human intervention or assistance. In order to make predictions without using explicit instructions, ML algorithms build a mathematical model based on sample data (training data). Afterward, the model quality is tested on the remaining datasets (test data) [32]. The key parameter in the learning process is the data, especially its quality and quantity. The bigger and cleaner dataset is, the better the output result is.

5.1. Types of Learning Algorithms

ML algorithms differ in the approach, type of input and output data and the problem to solve. Due to the type of provided dataset and the information they contain, it is possible to distinguish three major groups [33]:

- **Supervised learning** – build a mathematical model with the dataset which possesses both input and desired output information. It may be one or more input parameters, but there is always one output knows as a supervisory signal. The given dataset is called the training data and consists of training data points. Each training point must be represented as an array or the training data must be represented as a matrix. The supervised methods use the iterative optimization of an objective function to predict proper outputs. The optimal function should allow to properly predict the output for input data which are not included in the training dataset.
- **Unsupervised learning** – take a dataset that possesses only the inputs and tries to find the structure in the data for example grouping. Therefore, the algorithms learn from not labeled or categorized training data and identify commonalities in the dataset. The algorithms react and adjust the learning path based on the presence or absence of the identified commonalities.
- **Reinforcement learning** – the algorithms operate in a completely unknown environment without specific input or output data. The only information the machine receives is a so-called gain signal. This signal can be either positive (reward) or negative (punishment). The goal is to maximize the notion of cumulative reward.

5.2. Techniques in Supervised Learning

The supervised learning methods were chosen for creating both the formation classification and bit dull grading prediction models. In ML there is no one particular algorithm that is best for solving all problems. The exact technique should be chosen carefully based on the size and complexity of the dataset. Hence, the proper choice of the algorithm usually is complicated and depends on the trial and error method as well as user experience. Therefore, the supervised method can be divided into two subgroups [34]:

- **Classification algorithms** – allow assigning data points to appropriate categories based on one or more input variables. The process of assigning data into two categories is known as two-class classification, whereas if there are more labels the classification is called multiclass classification
- **Regression algorithms** – allow to estimate the relationship among variables and predict how the group of variables would behave in the future. It helps to understand how the criterion variable value changes while any of the independent variables are changed.

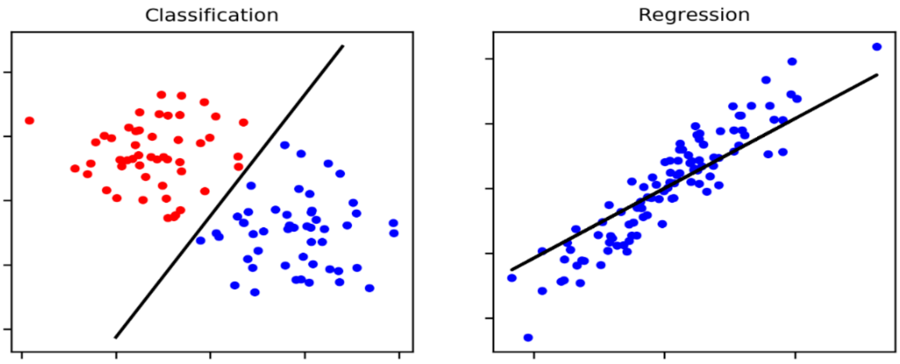


Figure 5.1 Difference between Classification and Regression algorithms [33].

5.3. Evaluating the Model

The next step after successfully running the model is its evaluation. This essential part of every machine learning project describes whether the algorithms have been able to properly predict the output from input data. The classification methods predict the class or the category to which the data belongs to. Hence, the classification algorithms accuracy is the most common method to describe model performance, however, it is not the only method which may judge the model [35].

- **Classification accuracy** – the most common method which uses the proportion of a number of correct prediction to the total number of input data points. The disadvantage of the method that it only works if each class are an equal number of samples.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions\ made} \tag{26}$$

- **Precision** – is the ratio of the true positives to the false positives and indicates the classifier's capability not to mark as positive observation point a point which is negative. The precision value range is between 0 – 1 and the greater the value, the better the model performance.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{27}$$

- **Recall** – is the ratio of the true positives to the false negatives and indicates the classifier's capability to find all positive observation points. The recall value range is between 0 – 1 and the greater the value, the better the model performance.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{28}$$

- **F1-score** – is the weighted average of the precision and recall where contribution of both parameters are equal. The f1-score value range is between 0 – 1 and the greater the value, the better the model performance.

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \tag{29}$$

- **Support** – it indicates how many times the true value occurred in each class

However, the regression models predict the quantity, so the model performance is evaluated as an error in made predictions [36].

- **Mean Absolute Error (MAE)** – it describes the average error magnitude, but don't describe the direction of the error (over or under predictions). It is average over the sample of the absolute differences between prediction and actual observation. It is calculated based on the following formula [37].

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_j - \widehat{y_i}\ | \tag{30}$$

  where:
  $n - the\ sample\ size$
  $|y_j - \widehat{y_i}| - absolute\ error$

- **Root Mean Squared Error (RMSE)** – it is the square root of the average squared differences between prediction and actual observation. It gives the gross perception of the error magnitude and may be calculated based on the given formula [38]

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_j - \widehat{y_i})^2} \tag{31}$$

  where:
  $n - the\ sample\ size$
  $(y_j - \widehat{y_i})^2 - squared\ differences$

- ***R^2*** **Metric** – it is also known as a coefficient of determination. It describes how the independent variables explain the variability in the dependent variable. The R^2 value is closer to 1, the better model accuracy is. It is calculated based on the formula [39].

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_j - \widehat{y_i})^2}{\frac{1}{n}\sum_{i=1}^{n}(y_j - \overline{y}i)^2} \tag{32}$$

  where:
  $n - the\ sample\ size$
  $SSE - squared\ error\ of\ the\ regression\ model$
  $SST - sum\ of\ squared\ errors\ of\ the\ baseline\ model$

5.4. Improving the Model

Sometimes, the model evaluation results are unsatisfactory. If the model training accuracy is low, it may indicate that the model configuration has not been able to correctly predict the output. In such situations, the simplest idea is to increase the number of data in the dataset. However, in this particular case, it has not been possible, so the algorithm tuning methods were applied.

One of the most common methods of improving model accuracy is hyper-parameter tuning. Hyper-parameter is the parameter which is controlled by the user of the ML model [40]. They have an impact on the model's parameters updating and model the learning process in the training phase. Hence, there is a possibility to set right hyper-parameter, the model would learn the most optimum weights for given training algorithms and dataset. The whole process may be done by using the scikit–learn library. Usually, it requires manual work to set the range of each hyper-parameter. The iteration over different hyper-parameters is quite time-consuming, especially while changing a lot of parameters simultaneously.

6. Formation Classification

The geological formation is the only factor which is truly independent of control. Each formation has its own characteristics that affect the drilling process. Hence, the aim of this study is to be able to classify the formations based on the recorded drilling parameters. Due to the lack of formation parameters, the model would predict only the formation name. The workflow for this study is presented below and will be thoroughly explained in further subchapters.

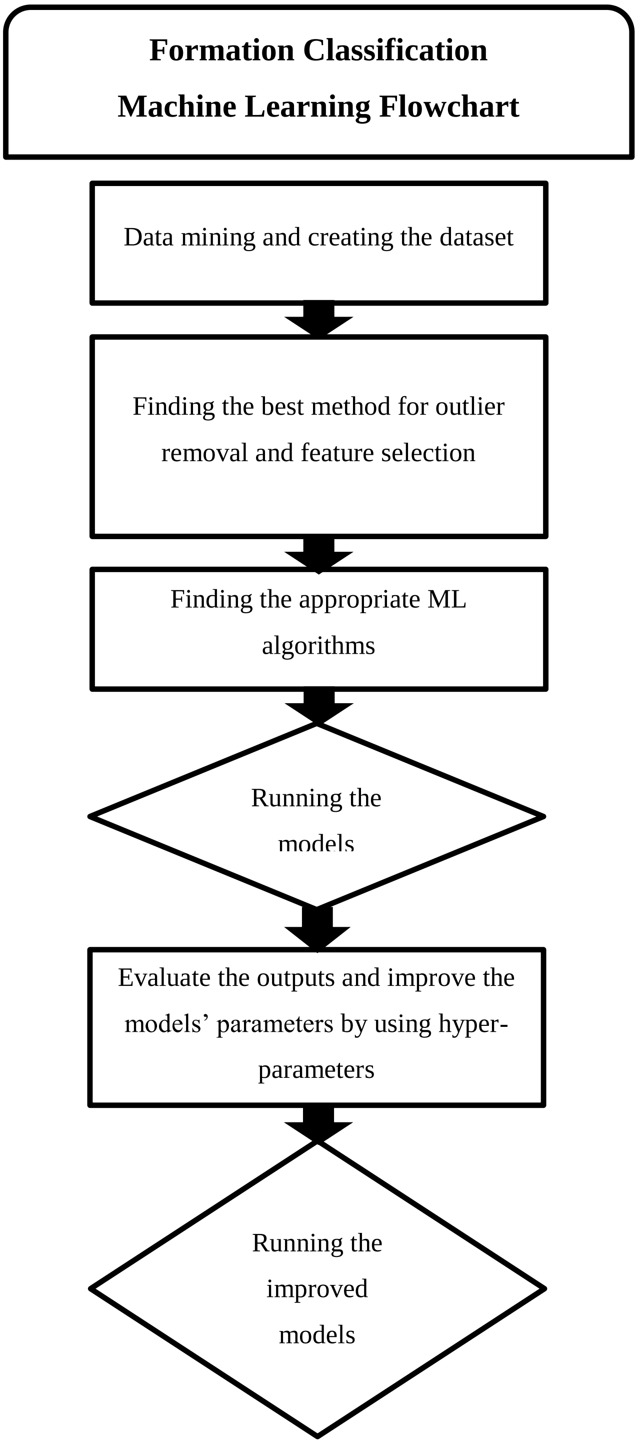

Formation Classification
Machine Learning Flowchart
Data mining and creating the dataset
Finding the best method for outlier removal and feature selection
Finding the appropriate ML algorithms
Running the models
Evaluate the outputs and improve the models' parameters by using hyper-parameters
Running the improved models

Figure 6.1 Formation classification machine learning flowchart.

6.1. Data Analytics

6.1.1. Data Preparation

The first step of the process has been the data extraction from the Volve field dataset. The observation points were divided into wells in which they belong to. There are 9 wells in total – F-1, F-4, F-5, F-7, F-9, F-10, F-11, F-14, F-15 and many of them are multilateral with additional wellbores. The data were extracted and stored in MS Excel sheets. Each well was divided into sections based on the bit runs. Due to the fact that each of the well contains not enough data for creating the robust machine learning model, all of the wells were merged together based on the well section size. Moreover, the intermediate and production well sections, 12 ¼” and 8 ½” are the most abundant in the data points and possess 1495 and 3626 observation points respectively. Hence, these sections were chosen for a further thorough investigation and based on them several machine learning models were created.

6.1.2. Outlier Removal

Having merged dataset, the second step has been the data quality improvement. Firstly, the whole dataset was plotted by using the pair plot from the seaborn library. It plots the pairwise relationship between all parameters in the dataset. Due to its size, the plot is attached as Appendix 7. The different colors in the pair plot indicate the different formation types. After knowing the relationships between parameters, the quality improvement was done by removing the outliers from the parameters. Each of the parameters in the dataset was plotted by using the boxplot. This method used the IQR formula to detect the outliers and plot the results in a readable manner. Some parameters from 8 ½” section were chosen from the dataset to illustrate how the dataset looked like before the outlier removal.

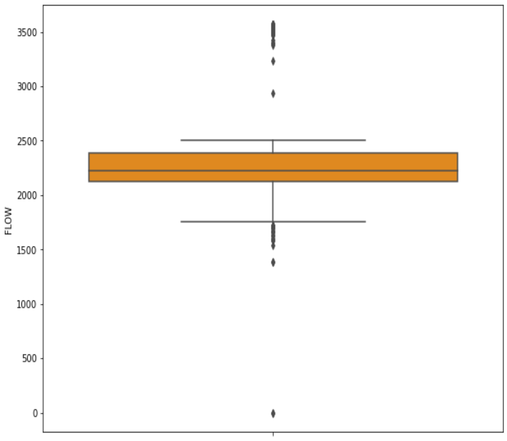


Chart 6.1 Flow In Rate boxplot.

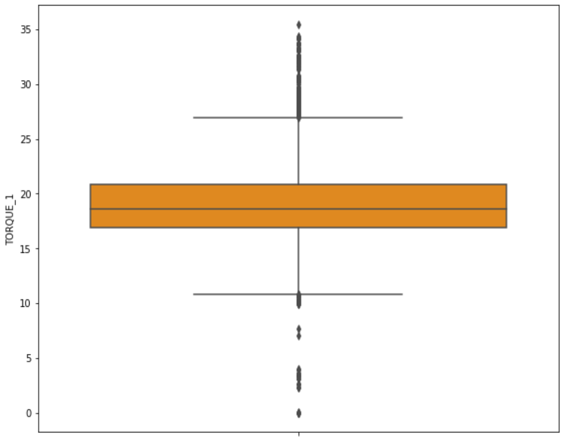


Chart 6.2 Torque boxplot.

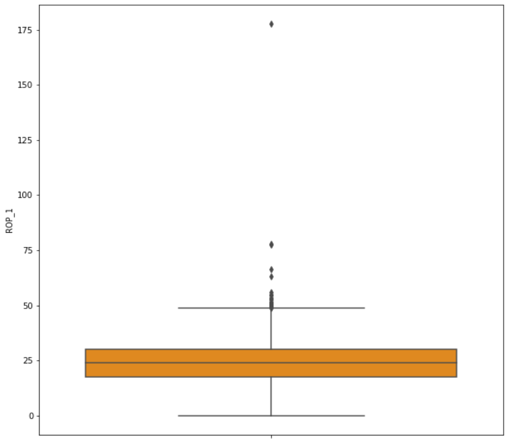


Chart 6.3 ROP boxplot.

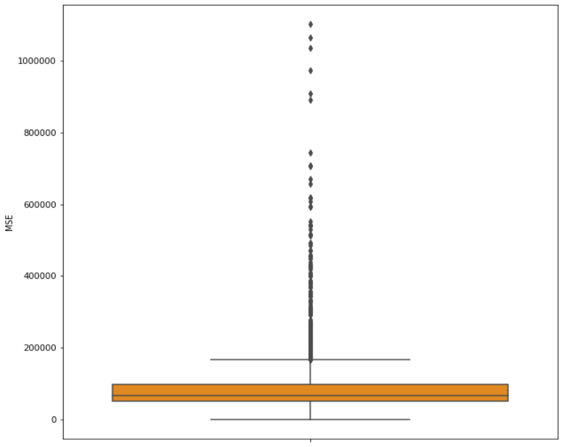


Chart 6.4 MSE boxplot.

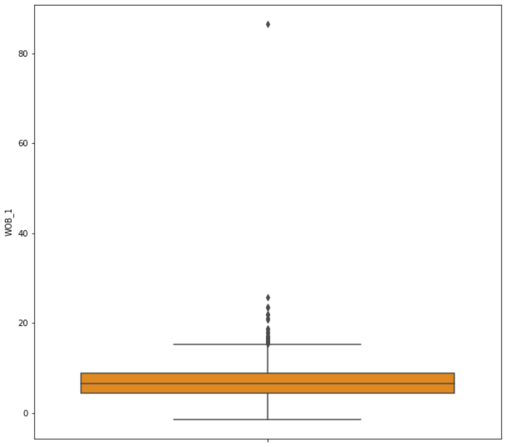


Chart 6.5 WOB boxplot.

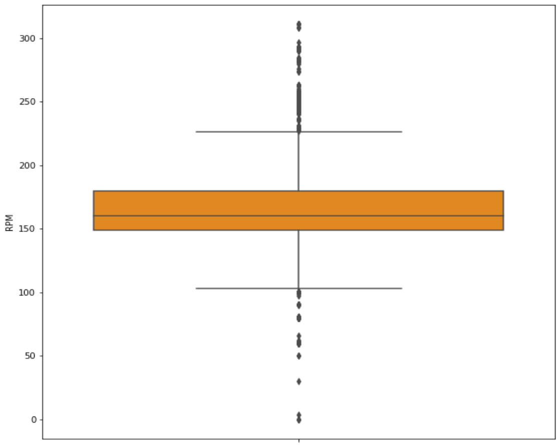


Chart 6.6 RPM boxplot.

The figures above clearly show that each parameter contains the outliers. In order to clean the dataset from the outliers, the Z - score and IQR method have been used for all of the features. The sample results for the 8 ½" section are shown below.

Samples = 3626	**IQR**	**Z -score**
Outliers	164	90
Samples after removal	3430	3536

Table 6.1 Z-score and IQR comparison for FLOW RATE IN – 8 ½" section.

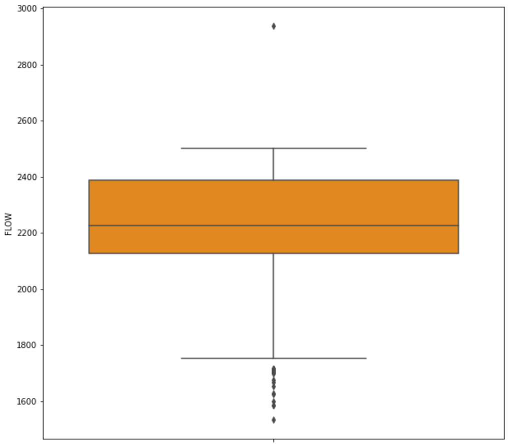


Chart 6.7 Flow In Rate after Z-score.

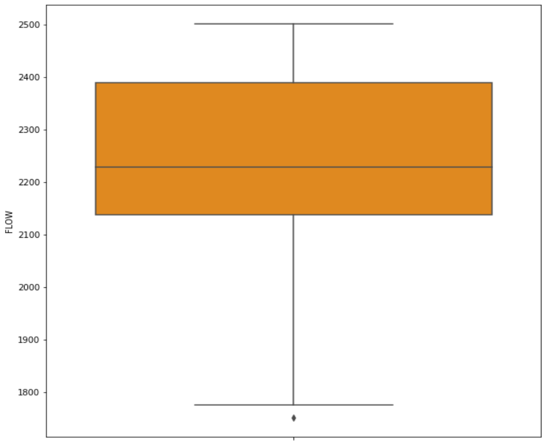


Chart 6.8 Flow In Rate after IQR.

Based on the given results the IQR method has been chosen for further outlier removal. MSE has been the only parameter in which the outliers were removed manually based on the scatter plot, due to its significant impact on formation classification. After removing the outliers form the parameters the 12 ¼” and 8 ½” possess 929 and 1969 observation points respectively.

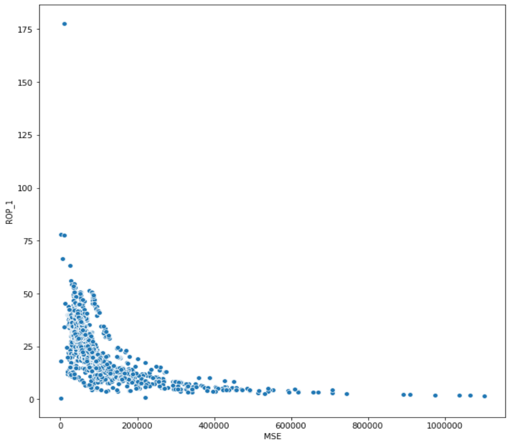


Chart 6.9 Plot before outlier removal.

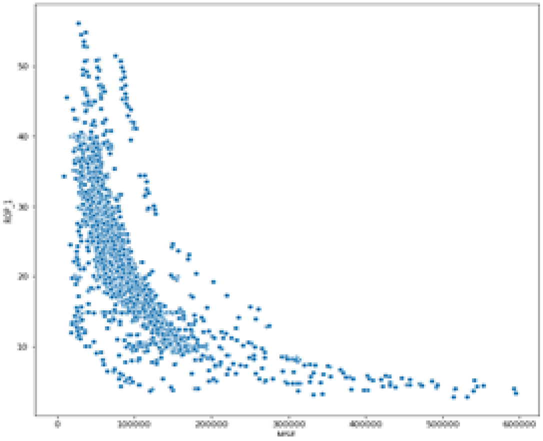


Chart 6.10 Plot after outlier removal.

6.1.3. Feature Selection

The next step after removing unnecessary data is to select the best features for the model. In order to find the most suitable features, the following techniques from the filter and wrapper methods have been tested: Correlation Matrix with Heatmap and Feature Importance. Due to the fact that ML can only understand the numbers, not strings, the data points such as lithology and mud type was changed to numbers. Every formation and mud type has its own number. The numbers where assigned in order in which the features occurred in the reports. The code for each method as well as the data conversion is attached in Appendix 8 and Appendix 9. The sample results for the 8 ½” section are shown in chart 6.12 and chart 6.13.

Geology	Number
Claystone	1
Sandstone	2
Siltstone	3
Tuff	4
Marl	5
Limestone	6
Coal	7

Table 6.2 Lithology conversion to numbers.

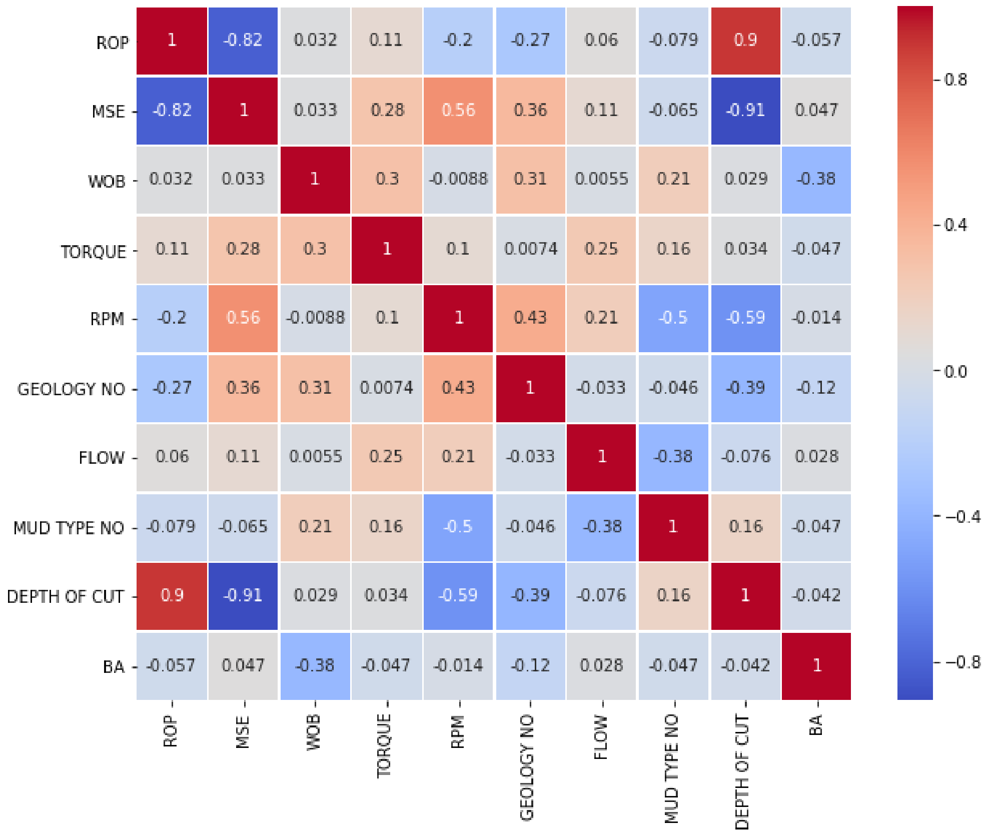


Chart 6.11 Correlation Matrix with Heatmap.

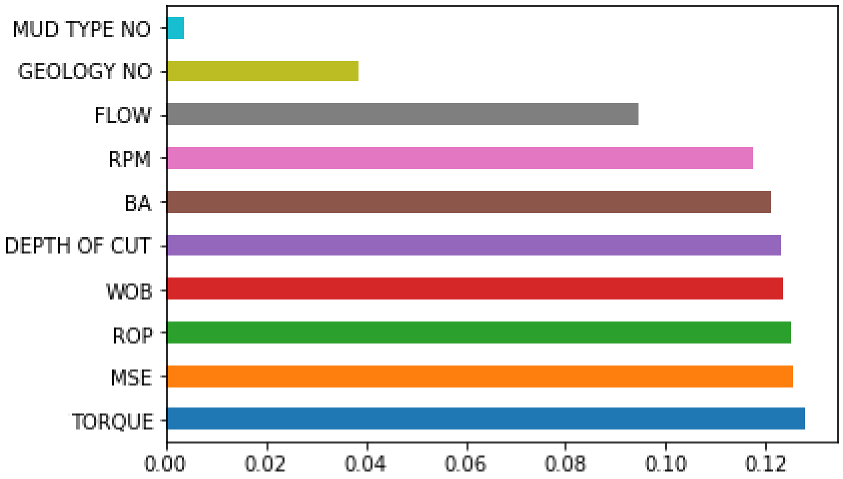


Chart 6.12 Feature Importance of Tree-Based Classifiers

The parameters showed above are the basic parameters measured during real-time operations (ROP, WOB, RPM, Torque, Flow Rate In, Mud Type No, Geology No) or they can be easily calculated from them (MSE, DoC, BA). Those parameters quite well describe the situation in the well, especially the interaction between the drillstring and the formation, so it is reasonable to use them as the input parameters for the ML models.

As may be seen in the above figures, each method gave different results. Hence, after analyzing the outcome and having the knowledge about the drilling operations, the following parameters were chosen as input variables to the formation classification model: MSE ($eq. 17$), Torque, DoC ($eq. 18$), BA ($eq. 19$), ROP, WOB, RPM and Flow Rate In.

6.2. Machine Learning Algorithms

6.2.1. Choosing the Models

Python has a dedicated library for ML, called scikit-learn. The library contains a wide range of algorithms, from classification techniques through regression and up to clustering [41]. The dataset consists of labeled data and the aim is to be able to properly select the formation. Such a problem is a perfect base for using the classification algorithms. The algorithms have been chosen based on the figure below.

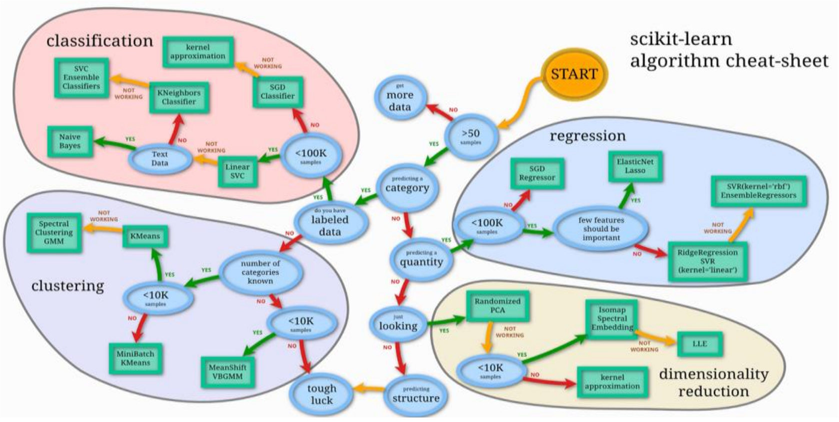


Figure 6.2 Scikit-learn algorithm cheat-sheet [39].

Datasets do not have any text data. According to the figure above, the most appropriate algorithm for such dataset is KNeighbors Classifier. However, the good idea is to be able to compare different models, so apart from KNeighbors Classifier, the Ensemble Classifiers such as Decision Tree, Random Forest, Extra Trees, and Gradient Boost Classifier have been used.

In all models, the test dataset size was set as 25% of the entire dataset. The observation points from the original dataset was chosen randomly, so the test dataset has points from majority of the bit runs. Moreover, each of the models possess the same test dataset in order to be able to see the differences between the models performance and debug the problems if they occur.

6.2.2. KNeighbors Classifier

The main idea of the algorithm is to find the k number of neighbor to the sample [42]. The algorithm should be able to correctly determine the most likely prediction. The k value is chosen by investigating the data. Generally, higher k value gives more accurate estimation due to better noise removal, however, k value in rage 3 – 10 is considered as a proper value.

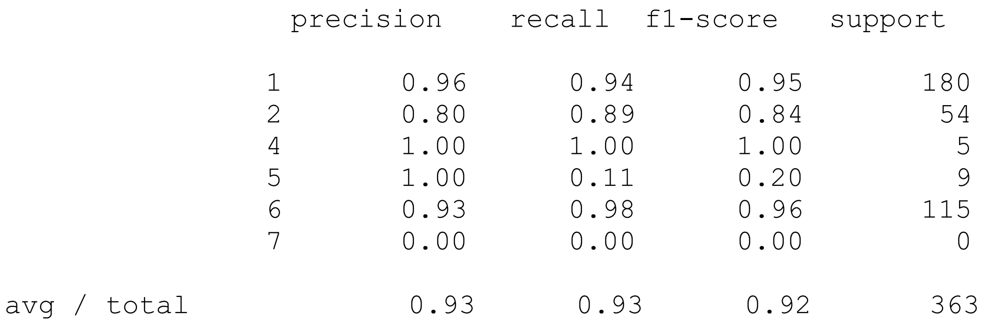
The first simulation was run with random parameters values and the code is attached in Appendix 10. The k value was selected to be equal to 5. The figures below show the results for both sections.

```
                precision     recall  f1-score    support

               1       0.57       0.61       0.59       153
               2       0.75       0.82       0.79       337
               3       0.35       0.23       0.28        35
               5       0.70       0.64       0.67       123
               6       0.84       0.78       0.81       246
               7       0.00       0.00       0.00         3

avg / total             0.72       0.73       0.72       897
```

Figure 6.3 Classification report – 8 ½” section.



```
                    precision    recall  f1-score   support

                1        0.96      0.94      0.95       180
                2        0.80      0.89      0.84        54
                4        1.00      1.00      1.00         5
                5        1.00      0.11      0.20         9
                6        0.93      0.98      0.96       115
                7        0.00      0.00      0.00         0

avg / total              0.93      0.93      0.92       363
```

Figure 6.4 Classification report – 12 ¼” section.

Section	Accuracy
8 ½	0.727
12 ¼	0.926

Table 6.3 The model accuracy for different sections.

The model evaluation shows that algorithms for section 12 ¼” are able to correctly determine the formation and improving the model by using the hyper-parameters is hardly possible. It may happen due to the fact that geology at this section is more uniform and there weren’t a lot of stringers and interbedded layers.

However, section 8 1/2” has a lot more complex and complicated geology, hence the accuracy is not as high as in the previous section. In order to increase the model performance, the code for estimating the best k value has been written. The code is attached in Appendix 11, while its result is shown below.

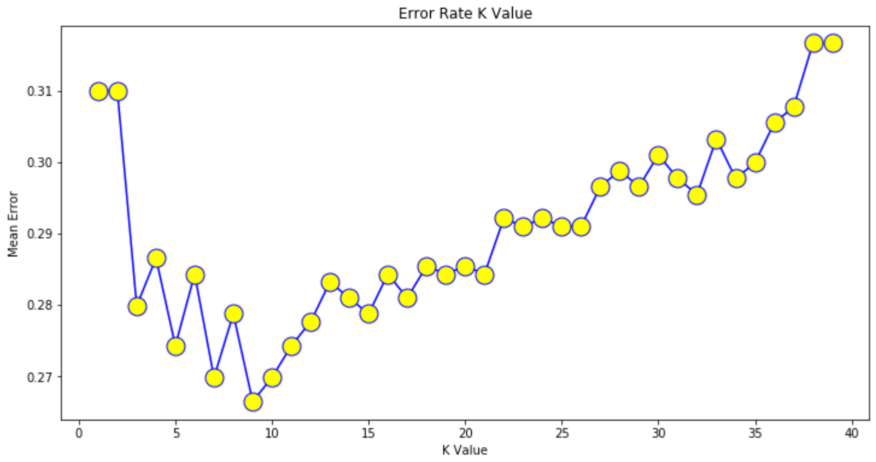


Chart 6.13 K-value predictions for 8 ½” section.

Based on the results shown on the figure above, the k value was set on 9. The results are presented below.

```
                precision    recall  f1-score   support

              1       0.61      0.63      0.58       153
              2       0.75      0.87      0.81       337
              3       0.45      0.24      0.32        35
              5       0.70      0.63      0.66       123
              6       0.86      0.80      0.82       246
              7       0.00      0.00      0.00         3

avg / total            0.73      0.74      0.73       897
```

Figure 6.5 Classification report after model improvement – 8 ½” section.

Section	Accuracy
8 ½	0.744

Table 6.4 The model accuracy after model improvement – 8 ½” section.

The new k value has nott significantly improved the model performance. It has only caused the minor change, however, the algorithm has looked promising. Such a situation may be caused by the algorithm inability to find proper neighbors and this may be caused by a high number of stingers and interbedded layers.

6.2.3. Decision Tree and Random Forest Classifier

The Decision Tree Classifier creates a tree-like model to predict possible paths of decisions and its consequences. Growing a tree takes into consideration the proper feature selection and deciding what conditions should be chosen for splitting dataset as well as at which condition the tree branch should stop growing. Generally, it may be shown as a response of two classes: Yes or No (1 or 0). This method is very simple and intuitive and can be combined with other decision techniques [42].

The next method is the Random Forest Algorithm. This is an ensemble of Decision Tree technique. The algorithm creates multiple decision trees and combines them together to get a correct and stable prediction. It only takes into consideration the random subset of features while splitting a node. The main difference between Decision Trees and Random Forest is that first algorithm will generate some rule while creating a tree, while second technique will randomly select data points and feature and afterward based on them build several decision trees and take the average of results. Another advantage of the Random Forest is that it prevents overfitting because the algorithm creates random subsets of data points [42].

The first simulation was run by using the Decision Tree algorithm with random parameters values. The test dataset size was set as 25% of the whole dataset and the parameter for both models was set as:

Parameter	**Value**
Criterion	gini
Max Depth	11
Max Features	auto
Max Leaf Nodes	None
Min Samples Leaf	1
Min Samples Split	15
Random State	42
Splitter	best

Table 6.5 The Decision Tree algorithm initial parameters.

The code is attached in Appendix 12 and figures below show the results for both sections.

```
             precision    recall  f1-score   support

          1       0.47      0.55      0.51       153
          2       0.73      0.69      0.71       337
          3       0.19      0.29      0.22        35
          5       0.56      0.57      0.56       123
          6       0.79      0.71      0.75       246
          7       0.00      0.00      0.00         3
avg / total       0.66      0.64      0.65       897
```

Figure 6.6 Decision Tree classification report – 8 ½” section.

```
             precision    recall  f1-score   support

          1       0.77      0.81      0.79       169
          2       0.61      0.58      0.60        60
          3       0.00      0.00      0.00         0
          4       0.50      0.50      0.50         8
          5       0.00      0.00      0.00         7
          6       0.76      0.76      0.76       120
          7       0.00      0.00      0.00         0

avg / total       0.72      0.73      0.73       364
```

Figure 6.7 Decision Tree classification report – 12 ¼” section.

Section	**Accuracy**
8 ½	0.648
12 ¼	0.734

Table 6.6 The model accuracy for different sections.

The results for both sections show that algorithms weren't able to determine the formation properly. The Grid Search CV library was used in order to find hyper-parameters and improve model performance. This library consists of many valuable tools which find the best parameters for the ML model. The written code is attached in Appendix 13, while the results after parameters improving are shown below.

```
                precision    recall  f1-score   support

              1       0.57      0.61      0.59       153
              2       0.75      0.78      0.77       337
              3       0.29      0.26      0.27        35
              5       0.73      0.62      0.67       123
              6       0.78      0.78      0.78       246
              7       0.00      0.00      0.00         3

avg / total           0.71      0.71      0.71       897
```

Figure 6.8 Decision Tree classification report after model improvement – 8 ½" section.

```
                precision    recall  f1-score   support

              1       0.85      0.79      0.82       174
              2       0.56      0.68      0.62        47
              4       0.75      0.75      0.75         4
              5       0.00      0.00      0.00         9
              6       0.79      0.86      0.82       133
              7       0.00      0.00      0.00         0

avg / total           0.77      0.78      0.77       367
```

Figure 6.9 Decision Tree classification report after model improvement – 12 ¼" section.

Section	Accuracy
8 ½	0.717
12 ¼	0.779

Table 6.7 The model accuracy after model improvement.

There was a successful improvement process in the 8 ½" section. The accuracy significantly increased and is almost similar to the KNeighbors method. However, there was no significant difference in the 12 ¼" section. The model accuracy slightly increased and is far lower than in KNeighbors technique.

The second simulation was run by using the Random Forest algorithm with random parameters values. The test dataset size was set as 25% of the whole dataset and the parameter for both models was set as:

Parameter	Value
Criterion	gini
Max Depth	11
Max Features	auto
Max Leaf Nodes	None
N-estimators	150

Table 6.8 Random Forest Algorithm initial parameters.

The code is attached in Appendix 14 and figures below show the results for both sections.

```
                 precision    recall  f1-score   support

              1       0.57      0.61      0.59       153
              2       0.75      0.78      0.77       337
              3       0.29      0.26      0.27        35
              5       0.73      0.62      0.67       123
              6       0.78      0.78      0.78       246
              7       0.00      0.00      0.00         3

avg / total          0.71      0.71      0.71       897
```

Figure 6.10 Random Forest classification report – 8 ½” section.

```
                 precision    recall  f1-score   support

             1       0.84      0.84      0.84      17411
             2         0.68      0.72      0.70        47
             4         0.00      0.00      0.00         4
             5         0.00      0.00      0.00         9
             6         0.84      0.90      0.87       133
             7         0.00      0.00      0.00         0

avg / total            0.81      0.82      0.80       367
```

Figure 6.11 Random Forest classification report – 12 ¼” section.

Section	Accuracy
8 ½	0.709
12 ¼	0.817

Table 6.9 The model accuracy for different sections.

The very similar algorithms as in the Decision Tree model were used to improve Random Forest model performance. The code for the model improvement is attached in Appendix 15.

```
                          precision    recall  f1-score   support

                        1       0.69      0.51      0.59       153
                        2       0.72      0.90      0.80       337
                        3       0.40      0.08      0.14        35
                        5       0.77      0.72      0.75       123
                        6       0.85      0.89      0.87       246
                        7       0.00      0.00      0.00         3

avg / total                     0.74      0.76      0.74       897
```

Figure 6.12 Random Forest classification report after model improvement – 8 ½” section.

```
                          precision    recall  f1-score   support

                        1       0.89      0.82      0.85       174
                        2       0.71      0.83      0.76        47
                        4       0.67      1.00      0.80         4
                        5       0.00      0.00      0.00         9
                        6       0.84      0.92      0.88       133
                        7       0.00      0.00      0.00         0


avg / total                     0.82      0.84      0.83       367
```

Figure 6.13 Random Forest classification report after model improvement – 12 ¼” section.

In both sections, the model performance was slightly improved. The 8 ½” section model has slightly better accuracy than the KNeighbors Classifier and the 12 ¼” section model results are between the Decision Tree and KNeighbours Classifier results.

Section	Accuracy
8 ½	0.757
12 ¼	0.841

Table 6.10 The model accuracy after model improvement.

6.2.4. Gradient Boost Classifier

The Gradient Boost Classifier is an example of the ensemble method. The key element to understand the algorithm is to know how the boosting actually works. Generally, by boosting the weak learners are converted into strong learners. Each new tree is fit on a modified version of the original dataset. The training process begins by creating a decision tree in which each data point has an equal weight. After the evaluation of the first tree, the weights of observation points that are hard to classify are increased, while the weight of observation points that are easy to classify is lowered. Therefore, the next tree is grown on the weighted data [43].

The simulation was run by using parameters from the improved Random Forest model. The code is attached in Appendix 16.

```
                  precision    recall  f1-score   support

                1      0.64      0.60      0.62       131
                2      0.75      0.84      0.79       237
                3      0.33      0.21      0.26        29
                5      0.71      0.66      0.68        93
                6      0.89      0.90      0.90       224
                7      0.00      0.00      0.00         2

avg / total            0.75      0.76      0.75       716
```

Figure 6.14 Gradient Boost classification report – 8 ½” section.

```
                   precision  recall  f1-score   support

                1      0.81      0.85      0.83       130
                2      0.82      0.74      0.78        54
                3      0.00      0.00      0.00         1
                4      1.00      0.83      0.91         6
                5      1.00      0.25      0.40         4
                6      0.83      0.86      0.84        99
                7      0.00      0.00      0.00         0

avg / total           0.82      0.82      0.82       294
```

Figure 6.15 Gradient Boost classification report – 12 ¼” section.

Section	Accuracy
8 ½	0.761
12 ¼	0.819

Table 6.11 The model accuracy.

The results for 8 ½” section is the best model performance from all presented methods, however, the results between the different techniques are very small. The result for 12 ¼” section is only better than in Decision Tree technique.

7. Bit Dull Grading Prediction

The drill bit is one of the key components of the drillstring and drilling process. Therefore, it is important to know as much as possible about the bit state during drilling. So far, the most common information about bit state is the information after running string into the hole and after pulling the string out of the hole. Hence, based on the Volve data the data-driven approach was used to predict the bit dull grade. The ability to predict properly the bit state based on the dataset should be valuable information during the well planning process as well as during field operations. The workflow for this study is presented below and will be explained in upcoming subchapters.

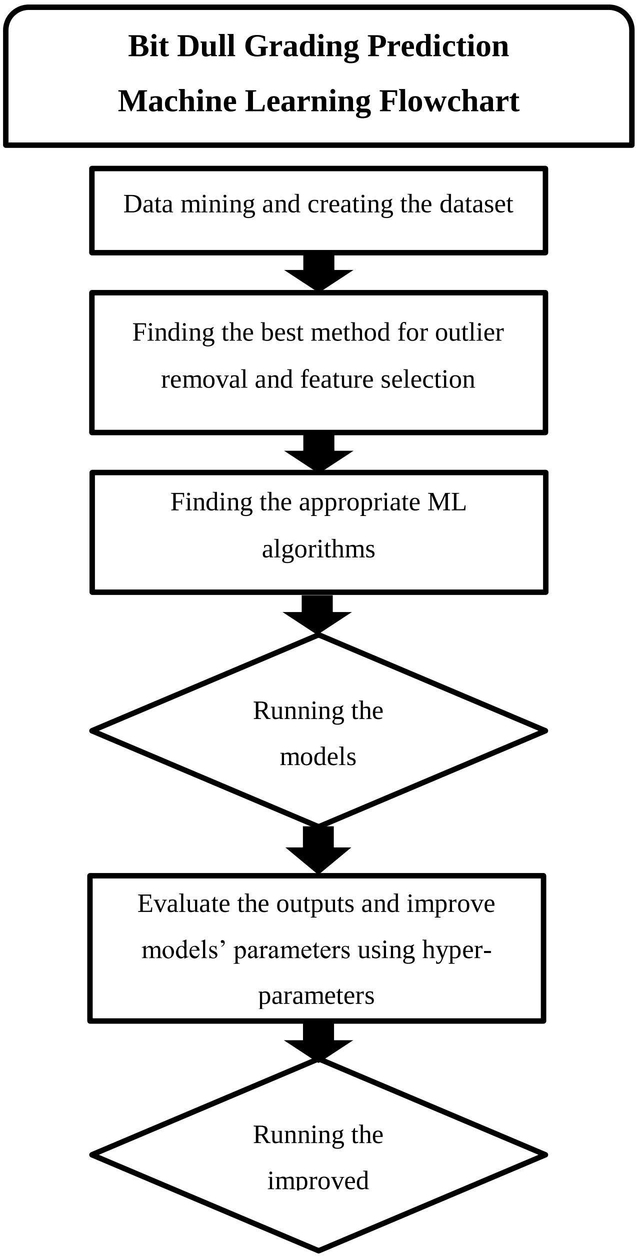

Bit Dull Grading Prediction
Machine Learning Flowchart
Data mining and creating the dataset
Finding the best method for outlier removal and feature selection
Finding the appropriate ML algorithms
Running the models
Evaluate the outputs and improve models' parameters using hyper-parameters
Running the improved

Figure 7.1 Bit dull grading prediction process flowchart.

7.1.Data Analytics

7.1.1. Data Preparation

Firstly, the data was extracted from the Volve field dataset and the same as in the formation classification part, the observation points were divided into the wells in which they belong to. The only difference between bit dull grading and formation classification dataset are the sections used for the model. In this model, sections 26”, 17 ½”, 12 ¼” and 8 ½” were used. All of the sections have enough observations points to be able to run the simulations. However, the section with the least observation points is the 26” section, due to the fact, that some of the wells were multilateral. Those wells usually have mother well up to 26” or 17 ½” section.

The major problem during the data mining process was to increase the number of the observation points. The ML algorithms work based on top-down approach. It means that each row of data must contain the parameter that describes the bit wear. However, in the drilling reports, the only known bit state is before running the drillstring into the hole and after pooling drillstring out of the hole. In order to increase the knowledge about the bit wear, the model presented in Chapter 2.3 and *Applied Drilling Engineering* textbook [2] was chosen to be implemented to increase the density of points. Such data transfer will significantly increase the number of points and it allows to obtain better model performance and accuracy.

7.1.2. Outlier Removal

The steps for outlier removal were very similar to the ones described in the formation classification case. Firstly, all dataset was plotted to get more insight into the correlations between each parameter. Having known such relations, it was possible to understand the parameters and divide them into two subsets. The first set was the parameters which could be removed automatically by using IQR method. Parameters like Bit Nozzle Velocity, Impact of Jet Nozzles on Hole Bottom, Cross Flow Velocity under the bit and Bit Aggressiveness were put in the first set. The computer could have done it blindly without losing any valuable observation points. The second set contained the remaining parameters which must have been dealt with more caution, in order not to lose valuable observation points which were classified by IQR methods as outliers.

Dataset size	26”	17 ½”	12 ¼”	8 ½”
Before IQR	1199	2869	1495	3626
After IQR	487	1006	524	1271

Table 7.1 The dataset size.

In order to understand the drilling data sample plots are shown below. The first plot shows the relation between BA and ROP. The outliers are clearly visible and using the IQR technique for BA parameter does not result in the loss of the valuable observation points.

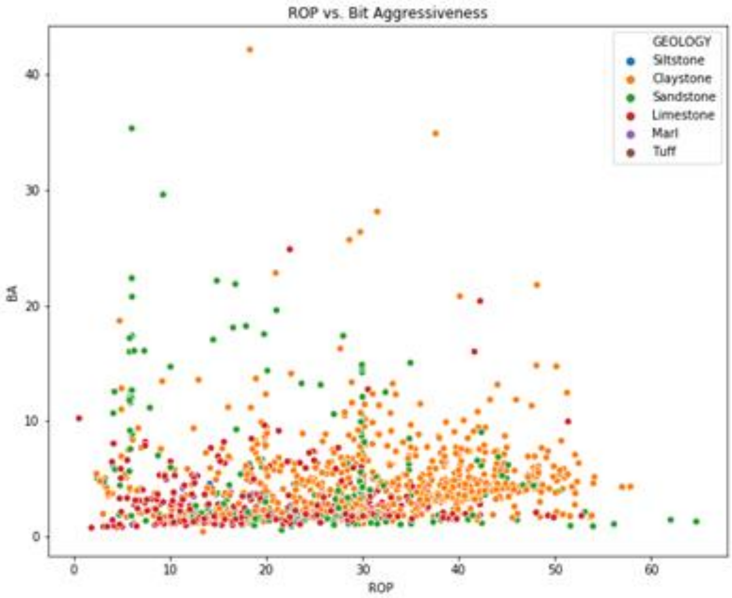


Chart 7.1 ROP vs. Bit Aggressiveness for section 12 ¼”: – before outlier removal.

However, while taking a look at the plot Torque vs. WOB it is clearly to detect the outliers, but for example, the outliers for the limestone may be valuable in order to predict the bit wear. Such a conclusion can be made based on the knowledge about the hardness and drillability of limestone as well as knowledge of the drilling parameters in different lithology. By manually removing outliers, the valuable data point is not deleted which would happen if the task is entrusted to the computer.

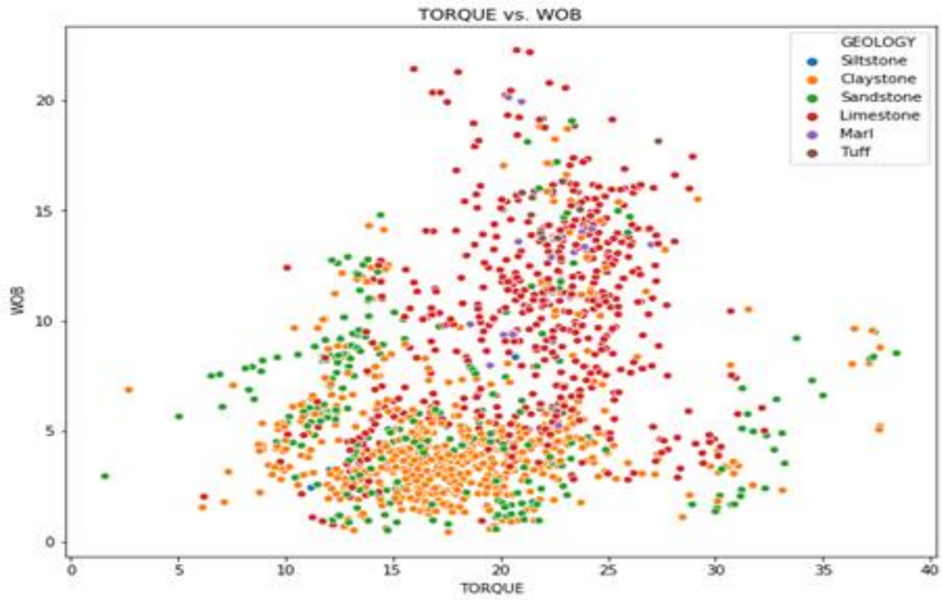


Chart 7.2 TORQUE vs WOB for section 12 ¼”: – before outlier removal.

7.1.3. Feature Selection

In this section, the steps are similar to the formation classification model as well. The most suitable features have been found by using the following: Correlation Matrix with Heatmap and Feature Importance. Due to the fact that ML can only understand the numbers, not strings, the data points such as Lithology and Mud Type was changed to numbers. The code for each method as well as the data conversion is attached in Appendix 8 and Appendix 9. The sample results for the 17 ½” section are shown below.

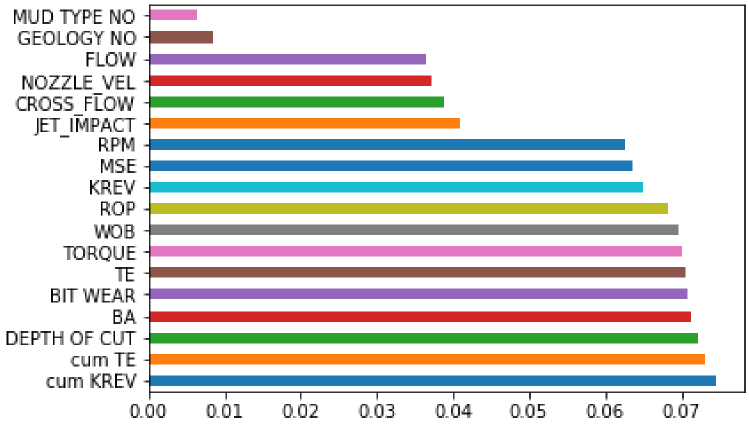
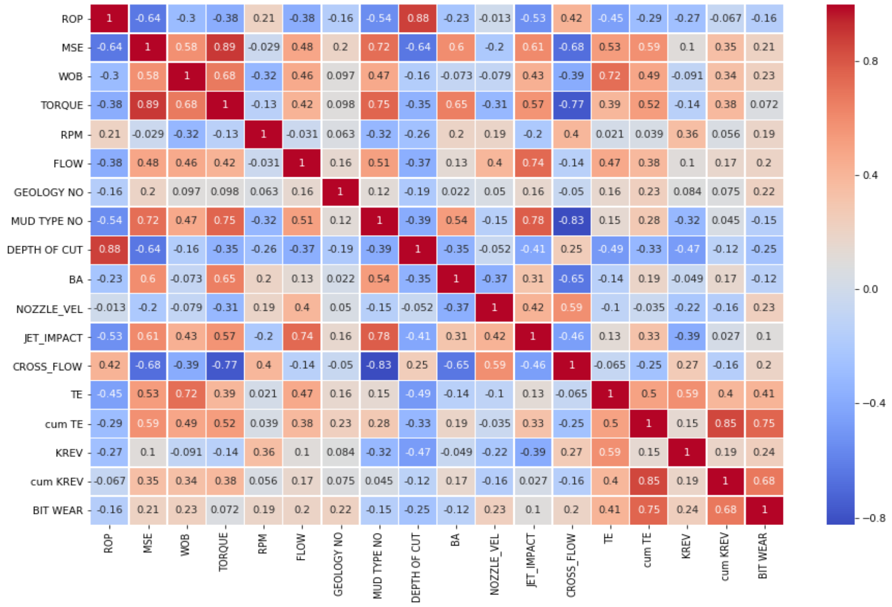


Chart 7.3 Feature Importance of Tree-Based Classifiers for section 17 ½”.



	ROP	MSE	WOB	TORQUE	RPM	FLOW	GEOLOGY NO	MUD TYPE NO	DEPTH OF CUT	BA	NOZZLE_VEL	JET_IMPACT	CROSS_FLOW	TE	cum TE	KREV	cum KREV	BIT WEAR
ROP	1	-0.64	-0.3	-0.38	0.21	-0.38	-0.16	-0.54	0.88	-0.23	-0.013	-0.53	0.42	-0.45	-0.29	-0.27	-0.067	-0.16
MSE	-0.64	1	0.58	0.89	-0.029	0.48	0.2	0.72	-0.64	0.6	-0.2	0.61	-0.68	0.53	0.59	0.1	0.35	0.21
WOB	-0.3	0.58	1	0.68	-0.32	0.46	0.097	0.47	-0.16	-0.073	-0.079	0.43	-0.39	0.72	0.49	-0.091	0.34	0.23
TORQUE	-0.38	0.89	0.68	1	-0.13	0.42	0.098	0.75	-0.35	0.65	-0.31	0.57	-0.77	0.39	0.52	-0.14	0.38	0.072
RPM	0.21	-0.029	-0.32	-0.13	1	-0.031	0.063	-0.32	-0.26	0.2	0.19	-0.2	0.4	0.021	0.039	0.36	0.056	0.19
FLOW	-0.38	0.48	0.46	0.42	-0.031	1	0.16	0.51	-0.37	0.13	0.4	0.74	-0.14	0.47	0.38	0.1	0.17	0.2
GEOLOGY NO	-0.16	0.2	0.097	0.098	0.063	0.16	1	0.12	-0.19	0.022	0.05	0.16	-0.05	0.16	0.23	0.084	0.075	0.22
MUD TYPE NO	-0.54	0.72	0.47	0.75	-0.32	0.51	0.12	1	-0.39	0.54	-0.15	0.78	-0.83	0.15	0.28	-0.32	0.045	-0.15
DEPTH OF CUT	0.88	-0.64	-0.16	-0.35	-0.26	-0.37	-0.19	-0.39	1	-0.35	-0.052	-0.41	0.25	-0.49	-0.33	-0.47	-0.12	-0.25
BA	-0.23	0.6	-0.073	0.65	0.2	0.13	0.022	0.54	-0.35	1	-0.37	0.31	-0.65	-0.14	0.19	-0.049	0.17	-0.12
NOZZLE_VEL	-0.013	-0.2	-0.079	-0.31	0.19	0.4	0.05	-0.15	-0.052	-0.37	1	0.42	0.59	-0.1	-0.035	-0.22	-0.16	0.23
JET_IMPACT	-0.53	0.61	0.43	0.57	-0.2	0.74	0.16	0.78	-0.41	0.31	0.42	1	-0.46	0.13	0.33	-0.39	0.027	0.1
CROSS_FLOW	0.42	-0.68	-0.39	-0.77	0.4	-0.14	-0.05	-0.83	0.25	-0.65	0.59	-0.46	1	-0.065	-0.25	0.27	-0.16	0.2
TE	-0.45	0.53	0.72	0.39	0.021	0.47	0.16	0.15	-0.49	-0.14	-0.1	0.13	-0.065	1	0.5	0.59	0.4	0.41
cum TE	-0.29	0.59	0.49	0.52	0.039	0.38	0.23	0.28	-0.33	0.19	-0.035	0.33	-0.25	0.5	1	0.15	0.85	0.75
KREV	-0.27	0.1	-0.091	-0.14	0.36	0.1	0.084	-0.32	-0.47	-0.049	-0.22	-0.39	0.27	0.59	0.15	1	0.19	0.24
cum KREV	-0.067	0.35	0.34	0.38	0.056	0.17	0.075	0.045	-0.12	0.17	-0.16	0.027	-0.16	0.4	0.85	0.19	1	0.68
BIT WEAR	-0.16	0.21	0.23	0.072	0.19	0.2	0.22	-0.15	-0.25	-0.12	0.23	0.1	0.2	0.41	0.75	0.24	0.68	1

Chart 7.4 Correlation Matrix with Heatmap for section 17 ½".

After the meetings with the industry representatives especially the bit design engineers, it turned out that cooling is one of the vital parameters for the PDC bit life prolongation. Having measured the flow rate in, it was possible to calculate the flow related parameters and use them as the input to the ML models. Similarly, the TE and KREV parameters were implemented. Having seen the chart 4.2 and 4.3, it was concluded that both parameters are valuable to evaluate and predict the bit dull grading. Moreover, the implementation of these parameters seems to be easy in the real-time operations and may give the valuable results, for example the rapid TE increase may indicate the harder stringer.

The parameters like ROP, RPM, WOB, Torque, MSE describes accurately the drilling process. The sudden increase in WOB may damage the bit and contribute to the faster bit worn-out. The higher torque values may tell about the poor bit condition, but relying solely on this particular parameter is not enough for proper bit evaluation. The ROP tells about the drilling performance and how fast the rock is being drilled. Based on the high ROP values it may be assumed, that bit works correctly and opposite if the ROP values are low. The MSE informs how much energy is used to drill the volume of rock. The MSE peak may indicate the harder stringers which usually causes the extensive bit wear.

Having got the understanding of the drilling process, its complexity and based on the obtained results from the feature selection methods, the following parameters were chosen to be input parameters for model: MSE $(eq. 17)$, DoC $(eq. 18)$, Torque, KREV $(eq. 21)$, TE $(eq. 20)$, cum TE, cum KREV, BA $(eq. 19)$, ROP, RPM, WOB, Nozzle Velocity $(eq. 22)$ and Cross Flow Velocity $(eq. 24)$.

7.2. Machine Learning Algorithms

7.2.1. Choosing the Models

Unlike the formation classification part, here there are not labeled data points. Hence, the regression part of scikit-learn library must have been used. Regression predictive modeling is the task of approximating a mapping function from input variables to the continuous output variable. The output variable is a real–value, for example, integer or float value. The selection of the proper algorithms was done by following the path presented in figure 6.2 in Chapter 6. The datasets size after outlier removal is presented in the table below.Based on figure 6.2 and table above, the following methods were selected to create the models: Lasso, Elastic Net, Ridge Regression, Decision Tree Regressor, Random Forest Regressor, and AdaBoost Regreesor.

7.2.2. Ridge Regression

This technique may be considered as an instance of the linear regression with regularization. It possesses an extra parameter α called tuning parameter in order to optimize the effect on multiple variable in linear regression. The higher value of the uning parameter, the higher residual sum of squares goes to zero. Generally, the tuning parameter describes the effect of the regression coefficients [44]. The model does not completely eliminate the coefficients and is relatively fast which is the main two differences between ridge regression and further described lasso technique.

The code for model improvement can be found in Appendix 17. The sample model parameters for section 12 ¼” are shown below.

Parameter	Value
Alpha	0.1
Max Iterations	5000
Solver	auto

Table 7.2 Ridge Regression Algorithm initial parameters.

The table 7.3, 7.4 and chart 7.5 shows the model evaluation as well as it shows the differences between the test and predicted values.

Evaluation	26”	17 ½”	12 ¼”	8 ½”
R^2	0.66	0.88	0.93	0.74
MAE	0.017	0.016	0.006	0.014
$RMSE$	0.025	0.022	0.008	0.018

Table 7.3 Ridge Regression Model Evaluation.

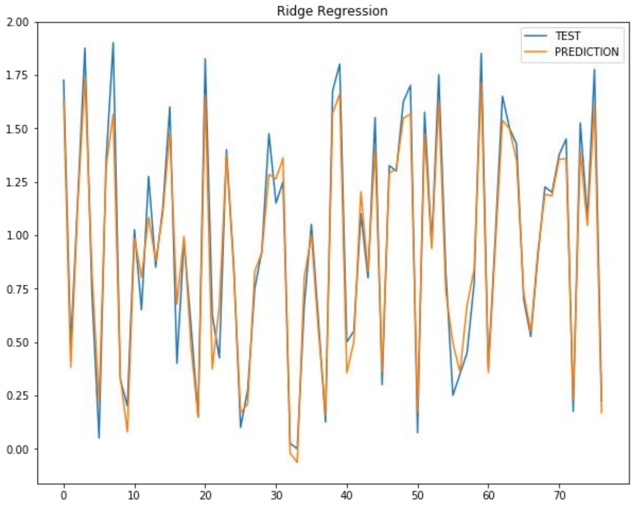


Chart 7.5 Ridge Regression test and predicted values comparison – 12 ¼ ”section.

Predicted Value	Test Value
0.10848912	0.108062
0.02743447	0.028112
0.07621225	0.077601

Table 7.4 Ridge Regression sample predicted and test values comparison – 12 ¼” section.

7.2.3. Lasso

The least absolute shrinkage and selection operator (Lasso) is an instance of the linear regression with regularization. It is very similar to earlier discussed ridge regression, however, it varies in the regularization values. Hence, it takes the absolute values of the sum of the regression coefficients and it may set the coefficients to zero in order to reduce the error completely. Simultaneously, the lasso method does the automatic variable selection for the model. Therefore, the ‘shrinkage’ feature comes from such behavior. The biggest advantage of the model is the ability to shrink the features. Therefore, the final calculations may use less (better quality) parameters than was used as input [45].

The code for model can be found in Appendix 18. The sample model parameters for section 12 ¼” are shown below. The table 7.6, 7.7 and chart 7.6 shows the model evaluation as well as it shows the differences between the test and predicted values.

Parameter	Value
Alpha	0.0001
Max Iterations	1000
Selection	random
Tolerance	0
Warm Start	True
Precompute	False

Table 7.5 Lasso Algorithm initial parameters.

The table 7.6, 7.7 and chart 7.6 shows the model evaluation as well as it shows the differences between the test and predicted values.

Evaluation	26”	17 ½”	12 ¼”	8 ½”
R^2	0.69	0.88	0.93	0.74
MAE	0.016	0.015	0.005	0.013
$RMSE$	0.022	0.023	0.007	0.017

Table 7.6 Lasso Model Evaluation.

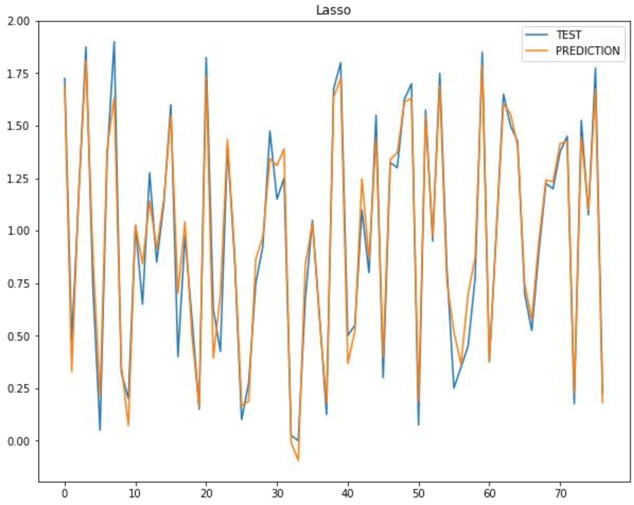


Chart 7.6 Lasso test and predicted values comparison – 12 ¼” section

Predicted Value	Test Value
0.1085012	0.108062
0.02756334	0.028112
0.07683221	0.077601

Table 7.7 Lasso sample predicted and test values comparison – 12 ¼” section.

7.2.4. Elastic Net

The Elastic Net is a hybrid technique that takes into account the penalties of the lasso and ridge regression methods. Generally, it means that it can effectively shrink coefficients as it happens in ridge regression as well as it may set some coefficients to zero as it occurs in the lasso technique. This technique tends to give results in between the ridge regression and lasso. The elastic net is the same as lasso when $\alpha = 1$ and as α goes to zero, the elastic net tends to approach the ridge regression [46].

The code for model can be found in Appendix 19. The sample model parameters for section 12 ¼” are shown below.

Parameter	Value
Alpha	0.0001
Max Iterations	1000
L1 ratio	0.1
Tolerance	0
Warm Start	True
Random State	10

Table 7.8 Elastic Net Algorithm initial parameters.

The table 7.9, 7.10 and chart 7.7 shows the model evaluation as well as it shows the differences between the test and predicted values.

Evaluation	26”	17 ½”	12 ¼”	8 ½”
R^2	0.64	0.90	0.95	0.74
MAE	0.017	0.016	0.005	0.012
$RMSE$	0.024	0.020	0.006	0.017

Table 7.9 Elastic Net Model Evaluation.

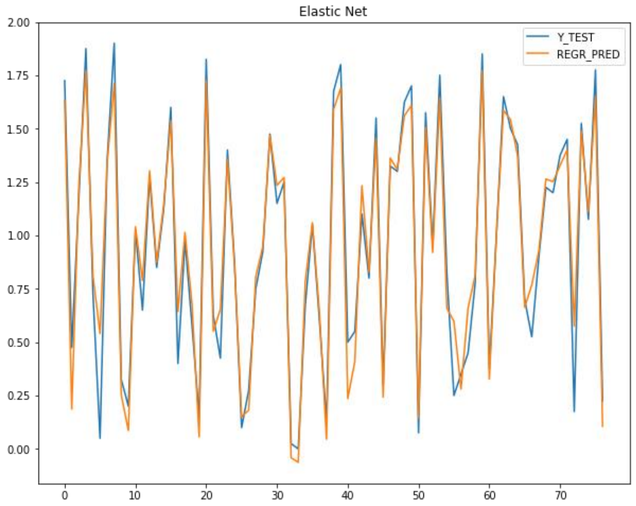


Chart 7.7 Elastic Net test and predicted values comparison –12 ¼” section.

Predicted Value	Test Value
0.10849812	0.108062
0.02733633	0.028112
0.07632561	0.077601

Table 7.10 Elastic Net sample predicted and test values comparison – 12 ¼” section.

7.2.5. Decision Tree and Random Forest Regressor

Generally, the Decision Tree Regressor works in a similar way as its equivalent to solve classification problems. However, there are some differences based on the structure of the dataset. The dataset for the regression tree does not have labeled data and the regression model is fitted to the target variable by using each of the preselected features. Afterward, for each preselected feature, the observation points are split at several split points. At each split point, the error between the predicted value and the actual values is squared to get a sum of squared errors (SSE). Then, each of the split points is compared with each other and the observation point with the lowest SSE is chosen to be a node point. This algorithm continues to recursively [47].

The next technique is the Random Forest Regressor. The principle is exactly the same as in the classification problem, however, it uses the Decision Tree Regressor instead of the Decision Tree Classifier to build the model and take the average results from build trees. The code for models can be found in Appendix 20 and 21.

Parameter	Value
Criterion	Auto
Max Depth	14
Max Features	auto
Min Samples Leaf	2
Min Samples Split	4
Presort	False
Splitter	best

Table 7.11 Decision Tree algorithm initial parameters.

The table 7.12, 7.13 and chart 7.8as well as the table 7.15, 7.16 and chart 7.9 shows Decision Tree and Random Forest models evaluation respectively. Figures show the differences between the test and predicted values.

Evaluation	26”	17 ½”	12 ¼”	8 ½”
R^2	0.97	0.99	0.99	0.99
MAE	0.006	0.001	0.001	0.001
$RMSE$	0.002	0.001	0.003	0.002

Table 7.12 Decision Tree Model Evaluation.

Predicted Value	Test Value
0.10849805	0.108062
0.02733744	0.028112
0.07632552	0.077601

Table 7.13 Decision Tree sample predicted and test values comparison – 12 ¼” section.

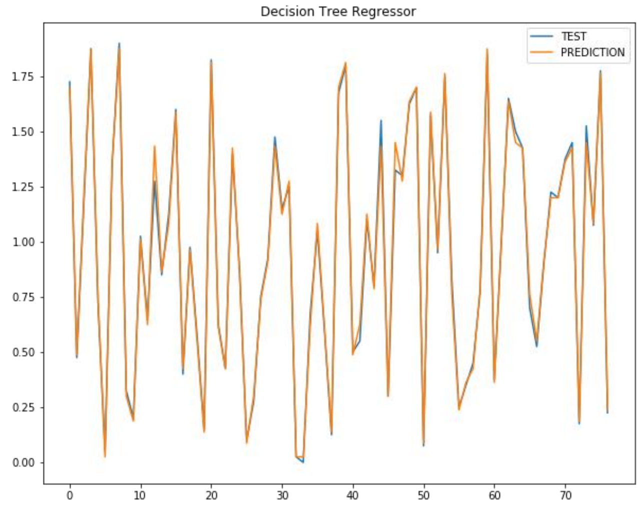


Chart 7.8 Decision Tree test and predicted values comparison – 12 ¼” section.

Parameter	Value
Criterion	Auto
Max Depth	13
Max Features	auto
Min Samples Leaf	2
Min Samples Split	4
Presort	False
N estimators	100

Table 7.14 Random Forest algorithm initial parameters.

Evaluation	26”	17 ½”	12 ¼”	8 ½”
R^2	0.96	0.98	0.97	0.97
MAE	0.008	0.001	0.004	0.006
$RMSE$	0.005	0.005	0.003	0.002

Table 7.15 Random Forest model evaluation.

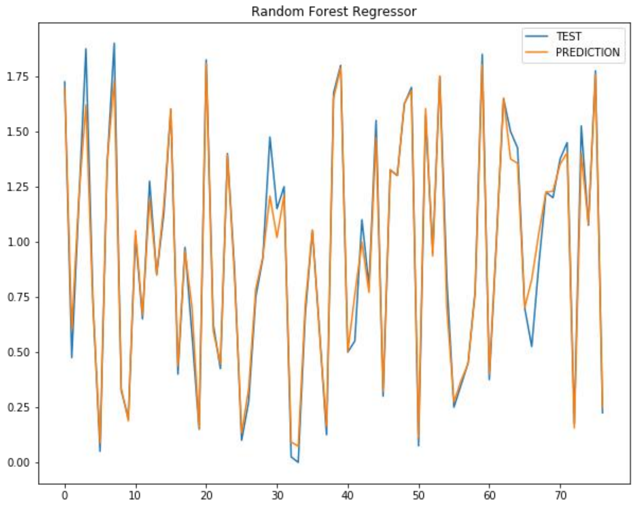


Chart 7.9 Random Forest test and predicted values comparison – 12 ¼” section.

Predicted Value	Test Value
0.10551514	0.108062
0.03780427	0.028112
0.07806395	0.077601

Table 7.16 Random Forest sample predicted and test values comparison – 12 ¼” section.

7.2.6. Ada Boost Regressor

The Ada Boost Regressor is an example of the boosting algorithm. As it was explained earlier, the boosting algorithms help to minimize or eliminate the randomness of the bagging techniques. The Ada Boost algorithm works based on the same principles described in Chapter 6 while explaining the Gradient Boost Classifier. However, there are two major differences. The first is the use of the Decision Tree Regressor instead of the Decision Tree Classifier. It is reasonable because the dataset has not got the labeled data and the goal is to predict the future value. The second difference is the process of choosing the proper weight of the decision trees. The Ada Boost technique chooses them probabilistically. This means that each of the decision trees has a certain probability and the sum of all decision trees probabilities is up to 1. The weight of observation points that are easy to classify is lowered while the weight of observation points hard to predict are increased [48]. The code for model can be found in Appendix 22. The sample model parameters were similar to the Decision Tree Regressor model. The table 7.17, 7.18 and chart 7.10 shows the model evaluation as well as it shows the differences between the test and predicted values.

Evaluation	26”	17 ½”	12 ¼”	8 ½”
R^2	0.99	0.99	0.99	0.99
MAE	0.002	0.004	0.003	0.003
$RMSE$	0.003	0.002	0.002	0.001

Table 7.17 AdaBoost model evaluation.

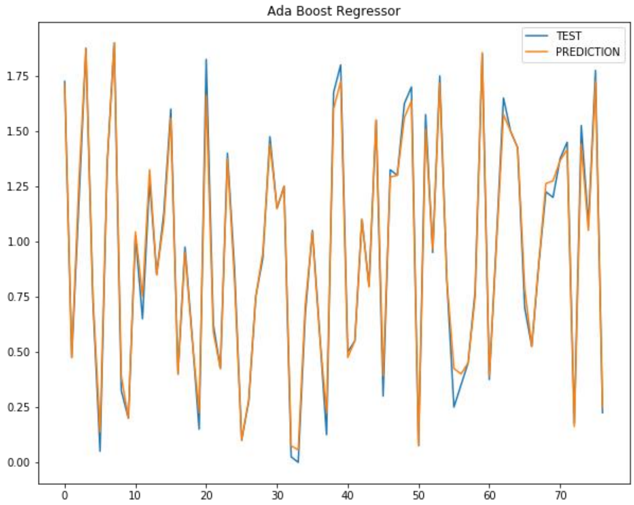


Chart 7.10 Ada Boost test and predicted values comparison – 12 ¼” section.

Predicted Value	Test Value
0.10714438	0.108062
0.28111710	0.028112
0.07125349	0.077601

Table 7.18 Ada Boost sample predicted and test values comparison – 12 ¼” section.

8. Conclusions

The Volve field dataset was analysed and pre-processed to create the Machine Learning algorithms. The work has shown that a data-driven approach may be successfully implemented on the field data. Moreover, it was shown that both classification and regression algorithms can have above average performance as well as they can accurately predict results. The most important part of the work was to understand the data, the physics which stays behind both the formation classification and bit dull prediction and prepare the robust datasets for ML algorithms.

The work could be divided into two parts. The first part was to obtain or calculate the most appropriate parameters which describes the actual conditions in the well in the best possible manner. The key parameter for the formation classification model was the MSE. Analysing the MSE values, it was relatively easy to distinguish between hard and soft formations in each section, for example between claystone or limestone. It was extremely beneficial in the 8 ½" sections, which was drilled through many geological faults. However, this parameter was not sufficient to properly evaluate the change between similar lithology.

The key parameter for the bit dull grading was the TE and KREV. The changes of these parameters, especially their cumulative values showed quite precisely where the bit work harder to drill the rock, thus where the bit worn-out faster. Even after thorough investigation it was impossible to combine the range of cumulative TE and KREV with the exact number of bit dull grading. However, having such a parameter would greatly simplify the proper bit selection.

The second part was associated with the data handling. Having such vast database, the crucial point was to remove outliers without losing valuable observation points. In order to do this, it is recommended to plot all of the parameters and see the relationship between them. Even if, the selected IQR method was classifying some observation points as outliers, it must have been seen if the qualified data did not represent any valuable information. Certainly, it had an impact on the model performance, usually, it has been decreasing the accuracy, but the valuable data were kept in the dataset.

In order to increase the model performance, the feature selection was conducted by using the two techniques – the Heat Map and the Tree Based Classifier. Based on them, the best available features were selected to be an input to the ML models.

8.1. Formation Classification

In order to predict the formation, the datasets were divided by the well section. Thanks to this assumption, the actual conditions were approximated more detailed. Such an approach was successful because merging the drilling and formation parameters from the entire well without having detailed information about the geology may have been misleading. Having data organized by well sections increases the chances of obtain the correct output. Most of the models were tuned by using the hyper-parameters technique and the table below shows the models accuracy before and after such improvement.

Model Evaluation	12 ¼”		8 ½”	
	Before	After	Before	After
$KNeighbours$	0.926	-	0.727	0.744
$Decision\ Tree$	0.734	0.779	0.648	0.717
$Random\ Forest$	0.817	0.841	0.709	0.757
$Gradient\ Boost$	0.819	-	0.761	-

Table 8.1 The classification algorithms accuracy comparison.

The hyper-parameters tuning significantly improved the models performance. Moreover, the model with the best accuracy for section 12 ¼” was KNeighbours Classifier and for section 8 ½” was Gradient Boost Classifier. It is hard to tell why one method was better than the other one. However, one of the reasons may be the difference in the classification pattern. All models have relatively similar performance, but it is clearly seen that the Decision Tree Classifier is the poorest model in both sections. Some methods were not upgraded by hyper-parameters, because the in these cases the difference was minimal between the version before and after testing with hyper-parameters ($|Accuracy| < 0.01$).

Generally, the table 8.1 shows that models performance was worse in the 8 ½” section. It is caused by the lithology complexity in that section. Usually, the 8 ½” section is drilled through many geological faults, so it is hardly possible to maintain the similar drilling parameters for each formation type. Therefore, such variations in drilling parameters may be an obstacle for the ML models and decline the correct formation prediction. The upper section – 12 ¼” is more lithologically coherent, hence the ML accuracy is higher.

Finally, the research was done by taking into consideration one filed. It turned out that the ML algorithms may correctly predict the lithology. The application of the formation classification algorithms has the potential to be implemented in the industry both in the well design and drilling phase. With the access to a larger database, the machine can be taught more cases, which will better predict the drilled formation. It may be extremely beneficial to run simulation with the designed drilling parameters during well design process and evaluate whether the chosen drilling parameters will ensure trouble-free drilling, avoiding washouts or hole stability problems.

In the real-time operations, the algorithms can be optimized to reach for example the highest ROP possible or to limit ROP in order to increase the hole stability. Also it may be used to evaluate whether the certain formation was reached in order to set casing shoe. Moreover, such algorithms may be used in order to increase the geosteering accuracy. However, the log and drilling data have not been combined in this thesis, but the idea may be used in further work. Not only may it increase the geosterring accuracy and the reservoir exposure, but also it may save a lot of time and money thanks to a better understanding of the formation-drillstring interaction.

8.2. Bit Dull Grading

The same approach was used in order to predict the bit of dull grading. The well data were divided based on the well section size. Such an approach enabled to differ between the roller cone and PDC bit. These bit types differ in working principles as well as in formulas to calculate the bit wear. The ML models where built by using similar steps as it was done in the formation classification case. Such an approach helped to build robust models which performance - R^2 is showed in the table below

Model Evaluation	**26”**	**17 ½”**	**12 ¼”**	**8 ½”**
Ridge Regression	0.66	0.88	0.93	0.74
Lasso	0.69	0.88	0.93	0.74
Elastic Net	0.64	0.90	0.95	0.74
Decision Tree	0.97	0.99	0.99	0.99
Random Forest	0.96	0.98	0.97	0.97
Ada Boost	0.99	0.99	0.99	0.99

Table 8.2 The regression algorithms accuracy comparison.

It is clearly seen that the ensemble algorithms (Decision Tree, Random Forest) and the boosting algorithm (Ada Boost) have significantly better performance than the other models

(Ridge Regression, Lasso, Elastic Net). The validation shows that ensemble methods, especially the AdaBoost Regressor gives better results, regardless the complexity of geology or the amount of data. It means that almost all of the points (even the anomalies) may be correctly predicted by the model and may be used for further development. On the other hand, the lower R^2 values ($R^2 < 0.7$) indicates that methods have the tendency to miss many important points, hence these methods should not be taken into consideration in future development. Moreover, the differences between each sections may result from the number of observation points and the complexity of geology in each section. As it was described in the previous subchapter, the 8 ½" was the most lithologically complicated section due to the many geological faults. On the other hand, the 26" section possessed the less complex lithology, but it has the least number of observation points which decrease the ML's ability to learn and correctly predict the output. Therefore, some of the methods had difficulties with correct prediction of bit grade in this section.

The obtained results show that it is possible to predict the bit wear based on the selected parameters. Having such information during the well design process would enable to reduce tripping time as well as eliminate the trial and error drill bit selection. It will ensure the more efficient and effective decision-making process. Due to the fact that the IADC code is still needed (hence the evaluation after the bit run), this approach may be used only while having a dataset of a fair size.

Despite the fact that both of the cases described in the thesis were independent work, it is possible to run them simultaneously. Having in mind the digitalisation process within industry, the both formation classification and bit dull grading predictions could be done in the real-time, helping the rig crew to meet the project requirements and avoid the unnecessary POOH due to excessive bit worn-out. However, such implementation requires larger database to be able to predict the output with higher accuracy.

In the future work, it would extremely beneficial to use the downhole parameters, for example, torque or RPM. Based on the comparison with surface parameters it should be possible to estimate whether the bit performs as expected and chose the appropriate moment to pull of the hole. Unfortunately, there was no possibility to get information about the formation parameters. However, in the future work, it would be beneficial to include the rock resistance, for example, the unconfined compressive strength and correlate it with the bit dullness. This approach combined with the drilling downhole data may lead to fruitful results.

References:

[1] Drill Bits, Baker Hughes a GE Company, https://www.bhge.com/upstream/drilling/drill-bits, [Accessed: 2019, Feb 02]

[2] A.T. Bourgoyne Jr., M.E. Chenevert, K.K. Millheim, F.S. Young Jr., Applied Drilling Engineering, SPE, pp. 190 – 240, 1986

[3] Drilling Bit Types and Drilling Bit Selection, drilingformulas.com, http://www.drillingformulas.com/drilling-bit-types-and-drilling-bit-selections/, [Accessed: 2019, Feb 03]

[4] W. Górski, Dobieranie narzędzi i osprzętu wiertniczego, Instytut technologii Eksploatacji – Państwowy Instytut Badawczy, Radom, 2007, p. 11

[5] Baker Hughes INTEQ, Drilling Engineering Workbook – A Distributed Learning Course, Houston, 1995, pp. 3.1 3.25

[6] W. H. Wamsley Jr., R. F. Mitchell. S Petroleum Engineering Handbook – Drilling Engineering, SPE, vol.2, pp. 221 – 264, 2007

[7] M. Varhaug, Schlumberger Defining Series – Bits, https://www.slb.com/-/media/Files/resources/oilfield_review/defining_series/Defining-Bits.pdf?la=en&hash=7AFD27FEAA283A428BEE202D460E458A604D0688, [Accessed: 2019, Feb 03]

[8] D. Y. McGehee, J.S. Dahlem, J.C. Gieck, B. Kost, D. Lafuze, C.H. Reinsvold, S.C. Steinke, The IADC Roller Bit Classification System, SPE/IADC 23937, New Orleans 1992

[9] D. Y. McGehee, J.S. Dahlem, J.C. Gieck, B. Kost, D. Lafuze, C.H. Reinsvold, S.C. Steinke, The IADC Roller Bit Dull Grading System, SPE/IADC 23938, New Orleans, 1992

[10] Products Category, Beste Bit, http://www.bestebit.com/product/geo-max-natural-diamond-bits/, [Accessed: 2019, Feb 03]

[11] H. Rabia, Well Engineering and Construction, Entrac Consulting, 2001, pp. 355-364

[12] B. D. Brandon, J. Cerkovnik, E. Koskie, B. B. Bayoud, F. Colston, R. I. Clayton, M. E. Anderson, K. T. Hollister, J. Senger, R. Niemi, Development of a New IADC Fixed Cutter Drill Bit Classification System, SPE/IADC 23940, New Orleans, 1992

[13] B. D. Brandon, J. Cerkovnik, E. Koskie, B. B. Bayoud, F. Colston, R. I. Clayton, M. E. Anderson, K. T. Hollister, J. Senger, R. Niemi, First Revision to the IADC Fixed Cutter Dull Grading System, SPE/IADC 23939, New Orleans, 1992

[14] B. Rashidi, G. Hareland, R. Nygaard, Real-Timne Drill Bit Wear Prediction by Combining Rock Energy and Drilling Strength Concepts, 117109, Abu Dhabi, 2008

[15] Z. Liu, C. Marland, D. Li, R. Samuel, An Analytical Model Coupled with Data Analytics to Estimate PDC Bit Wear, SPE 169451, 2014

[16] D. Belozerov, Drill Bits Optimization in the Ekofisk Overburden, University of Stavanger, 2015

[17] Disclosing All Volve Data, Equinor, https://www.equinor.com/en/news/14jun2018-disclosing-volve-data.html, [Accessed: 2019, Feb 06]

[18] A. Johansen, E. Kveinen, G. O. Byberg, K. A. Lehne, M. Skeide, S. Solymar, S. Ostensen, T. Nesse, T. H. Berge, Volve Field – Recommendation to Drill Well NO 15/9-F-7 and Well NO 15/9-F-9, Statoil, 2007

[19] Volve Field, Norwegian Petroleum, https://www.norskpetroleum.no/en/facts/field/volve/, [Accessed: 2019, Feb 07]

[20] S. Solymar, T. Odegard, M. Skeide, K. A. Lehne, E. L. Kristiansen, E. Kveinen, G. Byberg, P. I. Omdal, A. Johansen, Volve Field – Recommendation to Drill Well NO 15/9 F-4, Statoil, 2007

[21] J. VanderPlas, Python Data Science Handbook – Essential Tools for Working with Data, O'Reilly, 2017, pp. 5 – 10

[22]]A. Nagy, University of Stavanger, https://openlab.herokuapp.com/wells, [Accessed: 2019, Mai 06]

[23] R. Taele, The Concept of Specific Energy in Rock Drilling, International Journal of Rock Mechanics and Mining Sciences and Geomechanics, 1965, pp. 57 - 73

[24] K. Evans, S. C. Russell, Innovative Ability to Change Drilling responses of a PDC Bit at the Rigsite Using Interchangeable Depth-of-Cut Control Features, SPE 178808-MS, 2016

[25] R.C. Pessier, M. J. Fear, Quantifying Common Drilling Problems with mechanical Specific Energy and Bit-Specific Coefficient of Sliding Friction, SPE 24584, 1992

[26] Bit Calculations Reference, Beste Bit, http://www.bestebit.com/wp-content/uploads/2016/11/Bit-Calculations.pdf, [Accessed 2019, March 7]

[27] How to Use Statistics to Identify Outliers in Data, Machine Learning, Mastery, https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/, [Accessed: 2019, Feb 10]

[28] Interquartile Range, Statistic How To, https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/interquartile-range/, [Accessed: 2019, Feb 10]

[29] Z-Score: Definition, Formula and Calculation, Statistics How To, https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/z-score/, [Accessed: 2019, Feb 10]

[30] Why, How and When to apply Feature Selection, Towards Data Science, https://towardsdatascience.com/why-how-and-when-to-apply-feature-selection-e9c69adfabf2, [Accessed: 2019, Feb 12]

[31] An Introduction to Feature Selection, Machine Learning Mastery, https://machinelearningmastery.com/an-introduction-to-feature-selection/, [Accessed: 2019, Feb 12]

[32] A. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, O'Reilly, 2017, pp. 27 – 48

[33] Machine Learning, Wikipedia, https://en.wikipedia.org/wiki/Machine_learning, [Accessed: 2019, Feb 14]

[34] Supervised vs. Unsupervised Learning, Towards Data Science, https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d, [Accessed: 2019, Feb 14]

[35] A. Zheng, Evaluating Machine Learning Models – A Beginner's Guide to Key Concepts and Pitfalls, O'Reilly, 2015, pp. 7 - 36

[36] Metrics To Evaluate Machine Learning Algorithms in Python, Machine Learning Mastery, https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/, [Accessed: 2019, Feb 15]

[37] C. J. Willmott, K. Matsuura, Advantages of the Mean Absolute Error(MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance, Climate Research, 2005

[38] Mean Absolute Percentage Error, Statistics How To, https://www.statisticshowto.datasciencecentral.com/mean-absolute-percentage-error-mape/, [Accessed: 2019, Feb 29]

[39] Coefficient of Determination Explained, Towards Data Science, https://towardsdatascience.com/coefficient-of-determination-r-squared-explained-db32700d924e, [Accessed: 2019, Feb 15]

[40] F. Hutter, J. Lucke, L. Schmidt – Thieme, Beyond Manual Tuning of Hyperparameters, Springer, 2015

[41] Choosing the Right Estimator, scikit–learn, https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html, [Accessed: 2019, Feb 26]

[42] M. Kubat, An Introduction to Machine Learning, Springer, 2017, pp. 113 - 133

[43] Understanding Gradient Boosting Machines, Towards Data Science, https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab, [Accessed: 2019, Feb 26]

[44] A. K. Md. E. Saleh, M. Arashi, B. M. Golam Kibra, Theory of Ridge Regression Estimation with Applications, Wiley, 2019, pp. 1 – 39

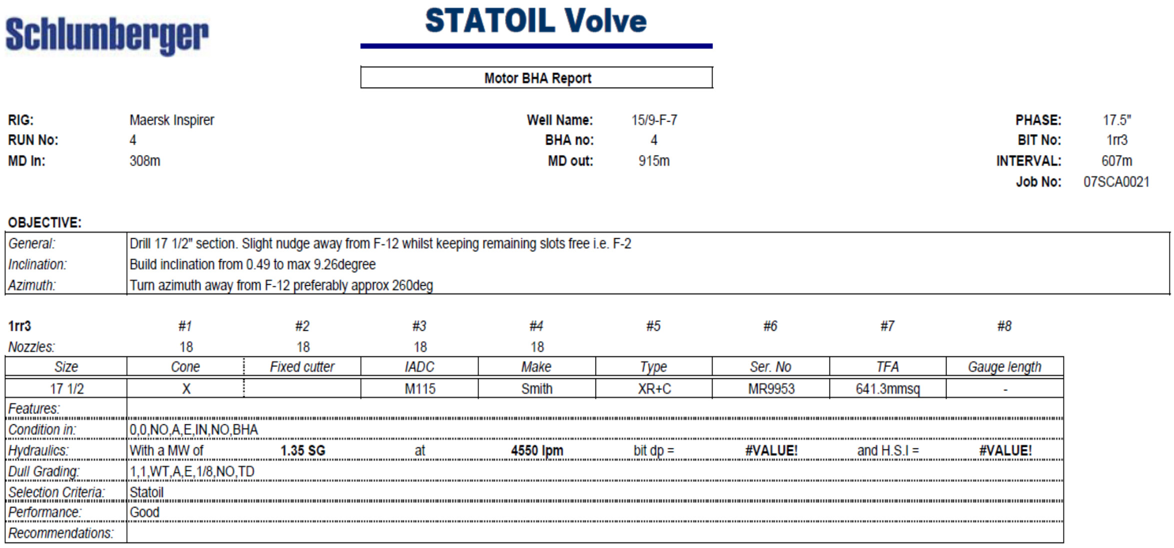
[45] T. Hastie, R. Tibshirani, M. Wainwright, Statistical Learning with Sparsity: The Lasso and Generalizations, CRC Press, 2015, pp. 7-24

[46] H. Zou, T. Hastie, Regularization and Variable Selection via the Elastic Net, Journal of the Royal Statistical Society, 2004

[47] B. Deshpande, Two Main Differences between Classification and Regression Trees, http://www.simafore.com/blog/bid/62482/2-main-differences-between-classification-and-regression-trees, [Accessed: 2019, March 10]

[48] M. Kubat, An Introduction to Machine Learning, Springer, 2017, pp. 179 - 189

Appendix 1. The Bit Dull Grading Chart (Statoil reports).



Schlumberger

STATOIL Volve

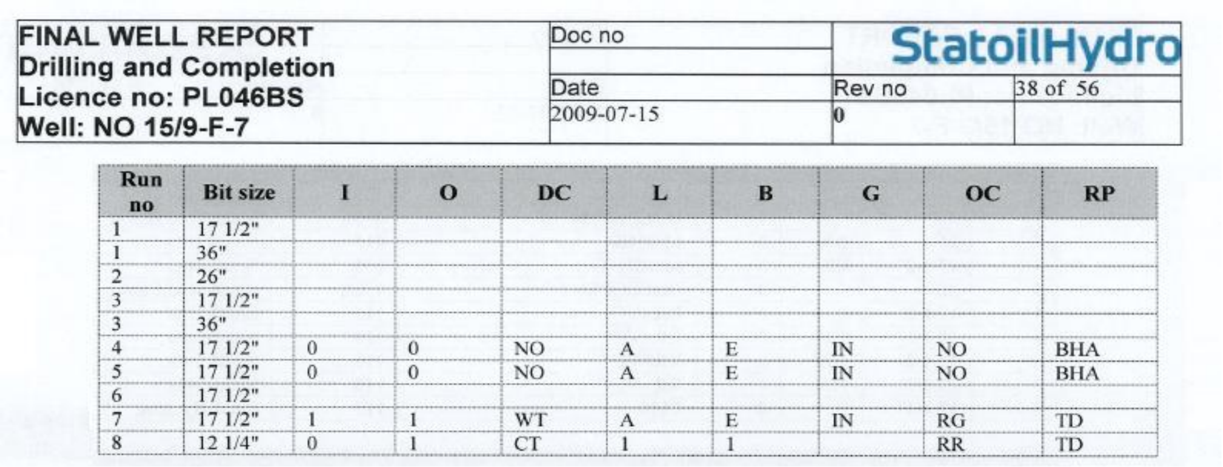
Motor BHA Report

RIG:	Maersk Inspirer	**Well Name:**	15/9-F-7	**PHASE:**	17.5"
RUN No:	4	**BHA no:**	4	**BIT No:**	1rr3
MD In:	308m	**MD out:**	915m	**INTERVAL:**	607m
				Job No:	07SCA0021

OBJECTIVE:

General:	Drill 17 1/2" section. Slight nudge away from F-12 whilst keeping remaining slots free i.e. F-2
Inclination:	Build inclination from 0.49 to max 9.26degree
Azimuth:	Turn azimuth away from F-12 preferably approx 260deg

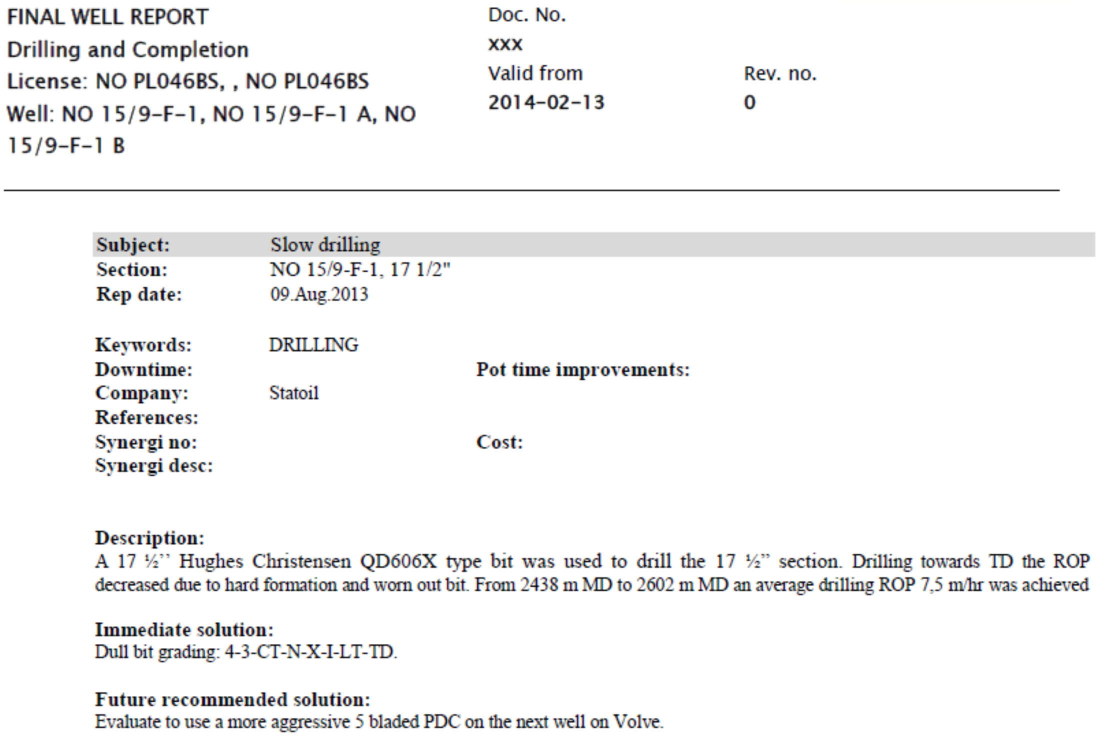
1rr3	*#1*	*#2*	*#3*	*#4*	*#5*	*#6*	*#7*	*#8*
Nozzles:	18	18	18	18				
Size	*Cone*	*Fixed cutter*	*IADC*	*Make*	*Type*	*Ser. No*	*TFA*	*Gauge length*
17 1/2	X		M115	Smith	XR+C	MR9953	641.3mmsq	-
Features:								
Condition in:	0,0,NO,A,E,IN,NO,BHA							
Hydraulics:	With a MW of	**1.35 SG**	at	**4550 lpm**	bit dp =	**#VALUE!**	and H.S.I =	**#VALUE!**
Dull Grading:	1,1,WT,A,E,1/8,NO,TD							
Selection Criteria:	Statoil							
Performance:	Good							
Recommendations:								



FINAL WELL REPORT Drilling and Completion Licence no: PL046BS Well: NO 15/9-F-7	Doc no	StatoilHydro	
	Date	Rev no	38 of 56
	2009-07-15	0	

Run no	Bit size	I	O	DC	L	B	G	OC	RP
1	17 1/2"								
1	36"								
2	26"								
3	17 1/2"								
3	36"								
4	17 1/2"	0	0	NO	A	E	IN	NO	BHA
5	17 1/2"	0	0	NO	A	E	IN	NO	BHA
6	17 1/2"								
7	17 1/2"	1	1	WT	A	E	IN	RG	TD
8	12 1/4"	0	1	CT	1	1		RR	TD

Appendix 2. The Drilling Problem Description (Statoil reports).



FINAL WELL REPORT
Drilling and Completion
License: NO PL046BS, , NO PL046BS
Well: NO 15/9-F-1, NO 15/9-F-1 A, NO 15/9-F-1 B

Doc. No.
xxx
Valid from
2014-02-13
Rev. no.
0

Subject:	Slow drilling		
Section:	NO 15/9-F-1, 17 1/2"		
Rep date:	09.Aug.2013		
Keywords:	DRILLING		
Downtime:		**Pot time improvements:**	
Company:	Statoil		
References:			
Synergi no:		**Cost:**	
Synergi desc:			

Description:
A 17 ½'' Hughes Christensen QD606X type bit was used to drill the 17 ½'' section. Drilling towards TD the ROP decreased due to hard formation and worn out bit. From 2438 m MD to 2602 m MD an average drilling ROP 7,5 m/hr was achieved

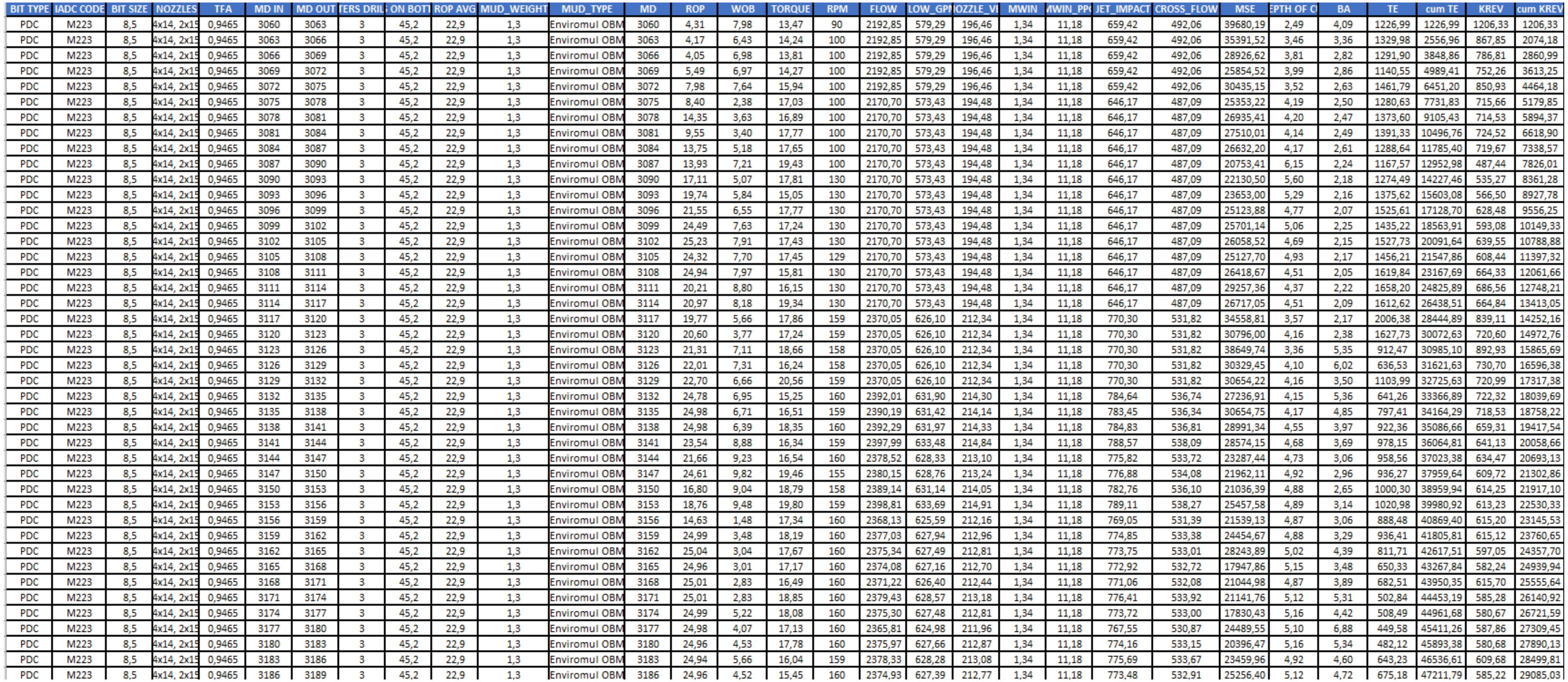
Immediate solution:
Dull bit grading: 4-3-CT-N-X-I-LT-TD.

Future recommended solution:
Evaluate to use a more aggressive 5 bladed PDC on the next well on Volve.

Appendix 3. The logs extraction (XML code).

```xml
<sources>
    <drillingData>
        <rootDirectory>C:\\Users\\jtfra\\Desktop\\Thesis\\Volve_Real_Time_DData\\WITSML Realtime drilling data\\Norway-Statoil-NO 15_$47$_9-F-11</rootDirectory>
        <drills>
            <drill>
                <rootDirectory>1</rootDirectory>
                <trajectory>trajectory\\1.xml</trajectory>
                <logs>
                    <log>log\\1\\3\\1\\00001.xml</log>
                    <log>log\\1\\3\\2\\00001.xml</log>
                    <log>log\\1\\3\\3\\00001.xml</log>
                    <log>log\\1\\3\\4\\00001.xml</log>
                </logs>
            </drill>
        </drills>
    </drillingData>
    <drillingData>
        <rootDirectory>C:\\Users\\jtfra\\Desktop\\Thesis\\Volve_Real_Time_DData\\WITSML Realtime drilling data\\Norway-Statoil-NO 15_$47$_9-F-11</rootDirectory>
        <drills>
            <drill>
                <rootDirectory>2</rootDirectory>
                <trajectory>trajectory\\1.xml</trajectory>
                <logs>
                    <log>log\\1\\6\\1\\00001.xml</log>
                    <log>log\\1\\6\\2\\00001.xml</log>
                    <log>log\\1\\6\\3\\00001.xml</log>
                </logs>
            </drill>
        </drills>
    </drillingData>
    <drillingData>
        <rootDirectory>C:\\Users\\jtfra\\Desktop\\Thesis\\Volve_Real_Time_DData\\WITSML Realtime drilling data\\Norway-Statoil-NO 15_$47$_9-F-11</rootDirectory>
        <drills>
            <drill>
                <rootDirectory>3</rootDirectory>
                <trajectory>trajectory\\1.xml</trajectory>
                <logs>
                    <log>log\\1\\5\\1\\00001.xml</log>
                    <log>log\\1\\5\\2\\00001.xml</log>
                    <log>log\\1\\5\\3\\00001.xml</log>
                </logs>
            </drill>
        </drills>
    </drillingData>
</sources>
```

Appendix 4. Part of the sample non – processed dataset (MS Excel)



BIT TYPE	IADC CODE	BIT SIZE	NOZZLES	TFA	MD IN	MD OUT	TERS DRIL	ON BOTT	ROP AVG	MUD_WEIGHT	MUD_TYPE	MD	ROP	WOB	TORQUE	RPM	FLOW	LOW_GPI	OZZLE_VI	MWIN	MWIN_PP	JET_IMPACT	CROSS_FLOW	MSE	EPTH OF C	BA	TE	cum TE	KREV	cum KREV
PDC	M223	8,5	4x14, 2x15	0,9465	3060	3063	3	45,2	22,9	1,3	Enviromul OBM	3060	4,31	7,98	13,47	90	2192,85	579,29	196,46	1,34	11,18	659,42	492,06	39680,19	2,49	4,09	1226,99	1226,99	1206,33	1206,33
PDC	M223	8,5	4x14, 2x15	0,9465	3063	3066	3	45,2	22,9	1,3	Enviromul OBM	3063	4,17	6,43	14,24	100	2192,85	579,29	196,46	1,34	11,18	659,42	492,06	35391,52	3,46	3,36	1329,98	2556,96	867,85	2074,18
PDC	M223	8,5	4x14, 2x15	0,9465	3066	3069	3	45,2	22,9	1,3	Enviromul OBM	3066	4,05	6,98	13,81	100	2192,85	579,29	196,46	1,34	11,18	659,42	492,06	28926,62	3,81	2,82	1291,90	3848,86	786,81	2860,99
PDC	M223	8,5	4x14, 2x15	0,9465	3069	3072	3	45,2	22,9	1,3	Enviromul OBM	3069	5,49	6,97	14,27	100	2192,85	579,29	196,46	1,34	11,18	659,42	492,06	25854,52	3,99	2,86	1140,55	4989,41	752,26	3613,25
PDC	M223	8,5	4x14, 2x15	0,9465	3072	3075	3	45,2	22,9	1,3	Enviromul OBM	3072	7,98	7,64	15,94	100	2192,85	579,29	196,46	1,34	11,18	659,42	492,06	30435,15	3,52	2,63	1461,79	6451,20	850,93	4464,18
PDC	M223	8,5	4x14, 2x15	0,9465	3075	3078	3	45,2	22,9	1,3	Enviromul OBM	3075	8,40	2,38	17,03	100	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	25353,22	4,19	2,50	1280,63	7731,83	715,66	5179,85
PDC	M223	8,5	4x14, 2x15	0,9465	3078	3081	3	45,2	22,9	1,3	Enviromul OBM	3078	14,35	3,63	16,89	100	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	26935,41	4,20	2,47	1373,60	9105,43	714,53	5894,37
PDC	M223	8,5	4x14, 2x15	0,9465	3081	3084	3	45,2	22,9	1,3	Enviromul OBM	3081	9,55	3,40	17,77	100	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	27510,01	4,14	2,49	1391,33	10496,76	724,52	6618,90
PDC	M223	8,5	4x14, 2x15	0,9465	3084	3087	3	45,2	22,9	1,3	Enviromul OBM	3084	13,75	5,18	17,65	100	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	26632,20	4,17	2,61	1288,64	11785,40	719,67	7338,57
PDC	M223	8,5	4x14, 2x15	0,9465	3087	3090	3	45,2	22,9	1,3	Enviromul OBM	3087	13,93	7,21	19,43	100	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	20753,41	6,15	2,24	1167,57	12952,98	487,44	7826,01
PDC	M223	8,5	4x14, 2x15	0,9465	3090	3093	3	45,2	22,9	1,3	Enviromul OBM	3090	17,11	5,07	17,81	130	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	22130,50	5,60	2,18	1274,49	14227,46	535,27	8361,28
PDC	M223	8,5	4x14, 2x15	0,9465	3093	3096	3	45,2	22,9	1,3	Enviromul OBM	3093	19,74	5,84	15,05	130	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	23653,00	5,29	2,16	1375,62	15603,08	566,50	8927,78
PDC	M223	8,5	4x14, 2x15	0,9465	3096	3099	3	45,2	22,9	1,3	Enviromul OBM	3096	21,55	6,55	17,77	130	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	25123,88	4,77	2,07	1525,61	17128,70	628,48	9556,25
PDC	M223	8,5	4x14, 2x15	0,9465	3099	3102	3	45,2	22,9	1,3	Enviromul OBM	3099	24,49	7,63	17,24	130	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	25701,14	5,06	2,25	1435,22	18563,91	593,08	10149,33
PDC	M223	8,5	4x14, 2x15	0,9465	3102	3105	3	45,2	22,9	1,3	Enviromul OBM	3102	25,23	7,91	17,43	130	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	26058,52	4,69	2,15	1527,73	20091,64	639,55	10788,88
PDC	M223	8,5	4x14, 2x15	0,9465	3105	3108	3	45,2	22,9	1,3	Enviromul OBM	3105	24,32	7,70	17,45	129	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	25127,70	4,93	2,17	1456,21	21547,86	608,44	11397,32
PDC	M223	8,5	4x14, 2x15	0,9465	3108	3111	3	45,2	22,9	1,3	Enviromul OBM	3108	24,94	7,97	15,81	130	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	26418,67	4,51	2,05	1619,84	23167,69	664,33	12061,66
PDC	M223	8,5	4x14, 2x15	0,9465	3111	3114	3	45,2	22,9	1,3	Enviromul OBM	3111	20,21	8,80	16,15	130	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	29257,36	4,37	2,22	1658,20	24825,89	686,56	12748,21
PDC	M223	8,5	4x14, 2x15	0,9465	3114	3117	3	45,2	22,9	1,3	Enviromul OBM	3114	20,97	8,18	19,34	130	2170,70	573,43	194,48	1,34	11,18	646,17	487,09	26717,05	4,51	2,09	1612,62	26438,51	664,84	13413,05
PDC	M223	8,5	4x14, 2x15	0,9465	3117	3120	3	45,2	22,9	1,3	Enviromul OBM	3117	19,77	5,66	17,86	159	2370,05	626,10	212,34	1,34	11,18	770,30	531,82	34558,81	3,57	2,17	2006,38	28444,89	839,11	14252,16
PDC	M223	8,5	4x14, 2x15	0,9465	3120	3123	3	45,2	22,9	1,3	Enviromul OBM	3120	20,60	3,77	17,24	159	2370,05	626,10	212,34	1,34	11,18	770,30	531,82	30796,00	4,16	2,38	1627,73	30072,63	720,60	14972,76
PDC	M223	8,5	4x14, 2x15	0,9465	3123	3126	3	45,2	22,9	1,3	Enviromul OBM	3123	21,31	7,11	18,66	158	2370,05	626,10	212,34	1,34	11,18	770,30	531,82	38649,74	3,36	5,35	912,47	30985,10	892,93	15865,69
PDC	M223	8,5	4x14, 2x15	0,9465	3126	3129	3	45,2	22,9	1,3	Enviromul OBM	3126	22,01	7,31	16,24	158	2370,05	626,10	212,34	1,34	11,18	770,30	531,82	30329,45	4,10	6,02	636,53	31621,63	730,70	16596,38
PDC	M223	8,5	4x14, 2x15	0,9465	3129	3132	3	45,2	22,9	1,3	Enviromul OBM	3129	22,70	6,66	20,56	159	2370,05	626,10	212,34	1,34	11,18	770,30	531,82	30654,22	4,16	3,50	1103,99	32725,63	720,99	17317,38
PDC	M223	8,5	4x14, 2x15	0,9465	3132	3135	3	45,2	22,9	1,3	Enviromul OBM	3132	24,78	6,95	15,25	160	2392,01	631,90	214,30	1,34	11,18	784,64	536,74	27236,91	4,15	5,36	641,26	33366,89	722,32	18039,69
PDC	M223	8,5	4x14, 2x15	0,9465	3135	3138	3	45,2	22,9	1,3	Enviromul OBM	3135	24,98	6,71	16,51	159	2390,19	631,42	214,14	1,34	11,18	783,45	536,34	30654,75	4,17	4,85	797,41	34164,29	718,53	18758,22
PDC	M223	8,5	4x14, 2x15	0,9465	3138	3141	3	45,2	22,9	1,3	Enviromul OBM	3138	24,98	6,39	18,35	160	2392,29	631,97	214,33	1,34	11,18	784,83	536,81	28991,34	4,55	3,97	922,36	35086,66	659,31	19417,54
PDC	M223	8,5	4x14, 2x15	0,9465	3141	3144	3	45,2	22,9	1,3	Enviromul OBM	3141	23,54	8,88	16,34	159	2397,99	633,48	214,84	1,34	11,18	788,57	538,09	28574,15	4,68	3,69	978,15	36064,81	641,13	20058,66
PDC	M223	8,5	4x14, 2x15	0,9465	3144	3147	3	45,2	22,9	1,3	Enviromul OBM	3144	21,66	9,23	16,54	160	2378,52	628,33	213,10	1,34	11,18	775,82	533,72	23287,44	4,73	3,06	958,56	37023,38	634,47	20693,13
PDC	M223	8,5	4x14, 2x15	0,9465	3147	3150	3	45,2	22,9	1,3	Enviromul OBM	3147	24,61	9,82	19,46	155	2380,15	628,76	213,24	1,34	11,18	776,88	534,08	21962,11	4,92	2,96	936,27	37959,64	609,72	21302,86
PDC	M223	8,5	4x14, 2x15	0,9465	3150	3153	3	45,2	22,9	1,3	Enviromul OBM	3150	16,80	9,04	18,79	158	2389,14	631,14	214,05	1,34	11,18	782,76	536,10	21036,39	4,88	2,65	1000,30	38959,94	614,25	21917,10
PDC	M223	8,5	4x14, 2x15	0,9465	3153	3156	3	45,2	22,9	1,3	Enviromul OBM	3153	18,76	9,48	19,80	159	2398,81	633,69	214,91	1,34	11,18	789,11	538,27	25457,58	4,89	3,14	1020,98	39980,92	613,23	22530,33
PDC	M223	8,5	4x14, 2x15	0,9465	3156	3159	3	45,2	22,9	1,3	Enviromul OBM	3156	14,63	1,48	17,34	160	2368,13	625,59	212,16	1,34	11,18	769,05	531,39	21539,13	4,87	3,06	888,48	40869,40	615,20	23145,53
PDC	M223	8,5	4x14, 2x15	0,9465	3159	3162	3	45,2	22,9	1,3	Enviromul OBM	3159	24,99	3,48	18,19	160	2377,03	627,94	212,96	1,34	11,18	774,85	533,38	24454,67	4,88	3,29	936,41	41805,81	615,12	23760,65
PDC	M223	8,5	4x14, 2x15	0,9465	3162	3165	3	45,2	22,9	1,3	Enviromul OBM	3162	25,04	3,04	17,67	160	2375,34	627,49	212,81	1,34	11,18	773,75	533,01	28243,89	5,02	4,39	811,71	42617,51	597,05	24357,70
PDC	M223	8,5	4x14, 2x15	0,9465	3165	3168	3	45,2	22,9	1,3	Enviromul OBM	3165	24,96	3,01	17,17	160	2374,08	627,16	212,70	1,34	11,18	772,92	532,72	17947,86	5,15	3,48	650,33	43267,84	582,24	24939,94
PDC	M223	8,5	4x14, 2x15	0,9465	3168	3171	3	45,2	22,9	1,3	Enviromul OBM	3168	25,01	2,83	16,49	160	2371,22	626,40	212,44	1,34	11,18	771,06	532,08	21044,98	4,87	3,89	682,51	43950,35	615,70	25555,64
PDC	M223	8,5	4x14, 2x15	0,9465	3171	3174	3	45,2	22,9	1,3	Enviromul OBM	3171	25,01	2,83	18,85	160	2379,43	628,57	213,18	1,34	11,18	776,41	533,92	21141,76	5,12	5,31	502,84	44453,19	585,28	26140,92
PDC	M223	8,5	4x14, 2x15	0,9465	3174	3177	3	45,2	22,9	1,3	Enviromul OBM	3174	24,99	5,22	18,08	160	2375,30	627,48	212,81	1,34	11,18	773,72	533,00	17830,43	5,16	4,42	508,49	44961,68	580,67	26721,59
PDC	M223	8,5	4x14, 2x15	0,9465	3177	3180	3	45,2	22,9	1,3	Enviromul OBM	3177	24,98	4,07	17,13	160	2365,81	624,98	211,96	1,34	11,18	767,55	530,87	24489,55	5,10	6,88	449,58	45411,26	587,86	27309,45
PDC	M223	8,5	4x14, 2x15	0,9465	3180	3183	3	45,2	22,9	1,3	Enviromul OBM	3180	24,96	4,53	17,78	160	2375,97	627,66	212,87	1,34	11,18	774,16	533,15	20396,47	5,16	5,34	482,12	45893,38	580,68	27890,13
PDC	M223	8,5	4x14, 2x15	0,9465	3183	3186	3	45,2	22,9	1,3	Enviromul OBM	3183	24,94	5,66	16,04	159	2378,33	628,28	213,08	1,34	11,18	775,69	533,67	23459,96	4,92	4,60	643,23	46536,61	609,68	28499,81
PDC	M223	8,5	4x14, 2x15	0,9465	3186	3189	3	45,2	22,9	1,3	Enviromul OBM	3186	24,96	4,52	15,45	160	2374,93	627,39	212,77	1,34	11,18	773,48	532,91	25256,40	5,12	4,72	675,18	47211,79	585,22	29085,03

Appendix 5. The IQR Calculation Method (Python code)

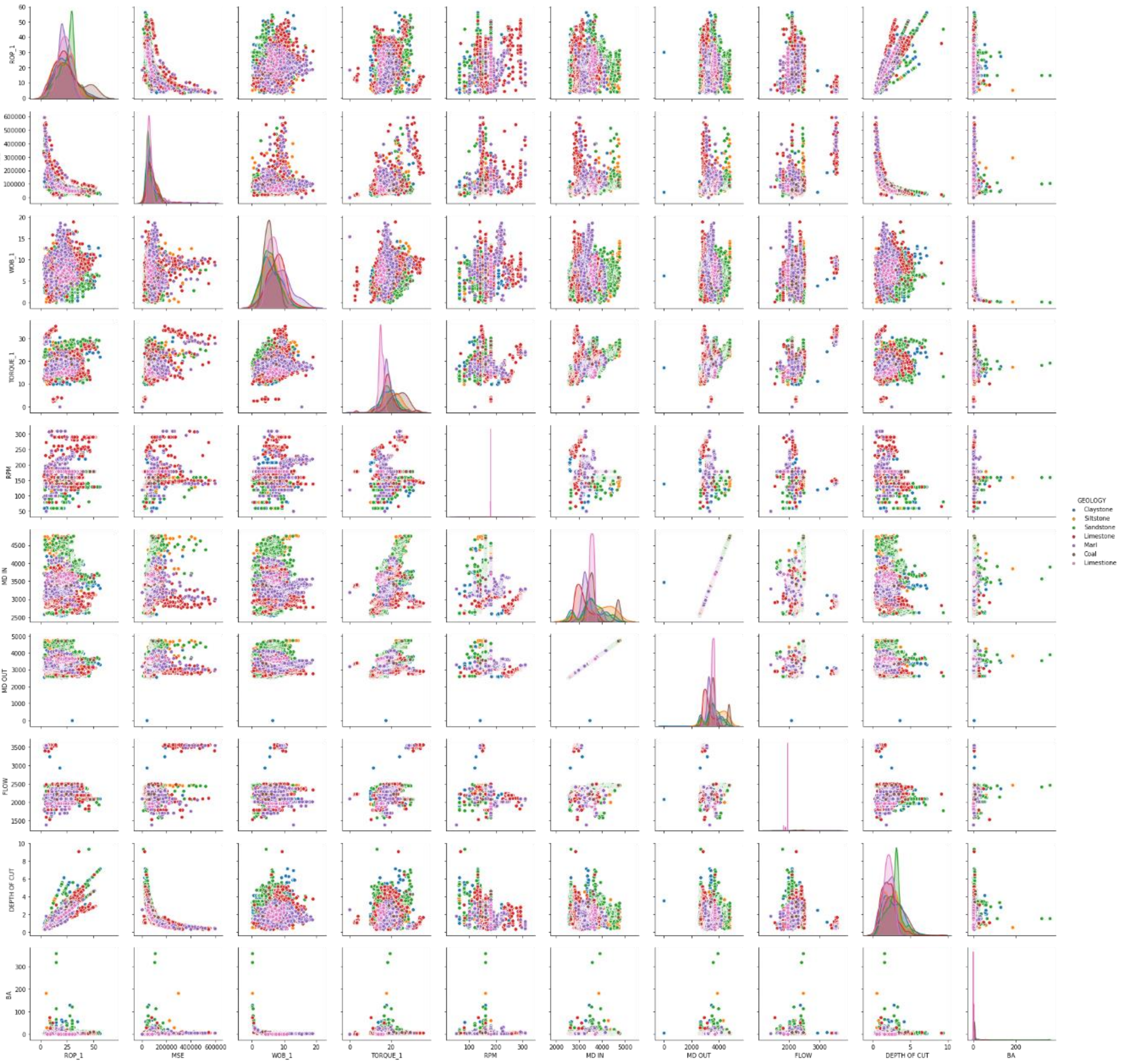
```python
def subset_by_iqr(dataset):
    q1 = dataset.quantile(0.25)
    q3 = dataset.quantile(0.75)
    iqr = q3 - q1

    filter = (dataset >= q1 - 1.5*iqr) & (dataset <= q3 + 1.5*iqr)
    return dataset.loc[filter]
```

Appendix 6. The Z-score Calculation Method (Python code)

```python
def calculate_z_score(dataset, treshold):
    dataset = dataset[((dataset - dataset.mean()) / dataset.std()).abs() < treshold]
    return dataset
```

Appendix 7. The Pairplot for 8 ½” Section (Python – Seaborn Library)



Appendix 8. The Transformation Parameters (Python code)

```python
def geology_value(c):
    geol = {'Claystone':1, 'Sandstone':2, 'Siltstone': 3, 'Tuff':4,  'Marl': 5, 'Limestone': 6, 'Coal': 7}
    return geol[c]

df['GEOLOGY NO'] = df['GEOLOGY'].apply(geology_value)

def mud_type_value(c):
    mud_type = {'Enviromul OBM':1, 'Standard OBM':2, 'OBM': 3}

    return mud_type[c]

df['MUD TYPE NO'] = df['MUD_TYPE'].apply(mud_type_value)
```

Appendix 9. The Feature Importace Selection (Python code)

```python
df_test = df[['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'GEOLOGY NO', 'FLOW', 'MWIN', 'MUD TYPE NO', 'DEPTH OF CUT', 'BA']]

X = df_test.iloc[:, 0:11]
y = df_test.iloc[:,1]

#data encoding
lab_enc = preprocessing.LabelEncoder()
y_encoded = lab_enc.fit_transform(y)

#univariate selection - mutual info regression
bestfeatures = SelectKBest(mutual_info_regression, k=11)
fit = bestfeatures.fit(X,y_encoded)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)

featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Paramters','Score']
print(featureScores.nlargest(11,'Score'))

#Tree Based Classifier
model = ExtraTreesClassifier()
model.fit(X,y_encoded)

print(model.feature_importances_)

feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(11).plot(kind='barh')
plt.show()

#correlation matrix with heatmap
corrmat = df_heat.corr()
top_corr_features = corrmat.index
sns.heatmap(df_heat[top_corr_features].corr(), annot = True, linewidth = 0.5, cmap = 'coolwarm')
```

Appendix 10. The KNeigbours Classifiers (Python code)

```python
#reading the datasets
X = df[['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'DEPTH OF CUT', 'FLOW', 'MWIN', 'BA']]
y = df['GEOLOGY NO']
#creating the train and test sets
X = preprocessing.scale(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)

#creating model
clf = neighbors.KNeighborsClassifier(n_neighbors=5)
clf.fit(X_train, y_train)
y_except = y_test
y_pred = clf.predict(X_test)
fpr, tpr, thresholds = metrics.roc_curve(y_except, y_pred, pos_label=2)

#showing results
print(metrics.classification_report(y_except, y_pred), clf.score(X_test, y_test))
print('confusion matrix', metrics.confusion_matrix(y_except, y_pred))
print('auc', metrics.auc(fpr, tpr))
```

Appendix 11. The KNeigbours Classifiers Model Improving (Python code)

```python
#creating a loop to fing best k value
error = []
for i in range(1, 40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)
    pred_i = knn.predict(X_test)
    error.append(np.mean(pred_i != y_test))

#creating a plot for showing the k values
plt.figure(figsize=(12, 6))
plt.plot(range(1, 40), error, color='blue', linestyle='-', marker='o',
         markerfacecolor='yellow', markersize=15)
plt.title('Error Rate K Value')
plt.xlabel('K Value')
plt.ylabel('Mean Error')
```

Appendix 12. The Decision Tree Classifier (Python code)

```python
#creating the dataset
X = df[['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'DEPTH OF CUT', 'FLOW', 'MWIN', 'BA']]
y = df['GEOLOGY NO']

#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)

#creating the model
clf = DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=19,
            max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=2, min_samples_split=9,
            min_weight_fraction_leaf=0.0, presort=False, random_state=42,
            splitter='best')

clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

#showing results
print(metrics.classification_report(y_test, y_pred))
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred, pos_label=2)
print('auc', metrics.auc(fpr, tpr))
```

Appendix 13. The Decision Tree Classifier Model Improving (Python code)

```python
#searching for best params
params_to_test = {'criterion': ['gini', 'entropy'],
                  'min_samples_split': range(2,20,1),
                  'min_samples_leaf': range(2, 20, 1),
                  'max_depth': range(1,20,1),
                  'splitter': ['best', 'random'],
                  'max_features': ['auto', 'sqrt', 'log2']}


#creating model
rf_model = DecisionTreeClassifier(random_state=42)
grid_search = GridSearchCV(rf_model, param_grid=params_to_test, cv=4, scoring='f1_macro', n_jobs=4)
grid_search.fit(X_train, y_train)

#showing results
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
print(best_params)
print(best_model)
```

Appendix 14. The Random Forest Classifier (Python code)

```python
#creating the dataset
X = df[['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'DEPTH OF CUT', 'FLOW', 'MWIN', 'BA']]
y = df['GEOLOGY NO']

#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)

#creating the model
clf = RandomForestClassifier(n_estimators=150, criterion = 'gini',
     max_depth = 18, min_samples_split = 2, random_state = 42, max_features = 'auto', splitter = 'best')

clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

#showing results
print(metrics.classification_report(y_test, y_pred))
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred, pos_label=2)
print('auc', metrics.auc(fpr, tpr))
```

Appendix 15. The Random Forest Classifier Model Improving (Python code)

```python
#searching for params
params_to_test = {'bootstrap': [True, False],
    'n_estimators':range(130, 150, 1), 'max_depth': range(5, 20, 1),
    'min_samples_split': range(2, 20, 1),'min_samples_leaf': range(2, 20, 1),

    'max_features': ['auto', 'sqrt', 'log2']}


#creating model
rf_model = RandomForestClassifier(random_state=42)
grid_search = GridSearchCV(rf_model, param_grid=params_to_test, cv=4, scoring='f1_macro', n_jobs=5)
grid_search.fit(X_train, y_train)

#showing results
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
print(best_params)
print(best_model)
```

Appendix 16. The Gradient Boost Classifier (Python code)

```python
#creating the dataset
X = df[['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'DEPTH OF CUT', 'FLOW', 'MWIN', 'BA']]
y = df['GEOLOGY NO']

#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

#creating the model
boost =GradientBoostingClassifier(
    learning_rate=0.99,
    n_estimators=150,
    max_depth =  17)
model = boost.fit(X_train, y_train)

y_pred = model.predict(X_test)
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred, pos_label=2)

#showing results
print(metrics.classification_report(y_test, y_pred))
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print('auc', metrics.auc(fpr, tpr))
```

Appendix 17. The Ridge Regressor (Python code)

```python
#creating the dataset
X = df['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'FLOW', 'DEPTH OF CUT','BA', 'NOZZLE_VEL', 'JET_IMPACT', 'CROSS_FLOW', 'TE', 'cum TE', 'KREV', 'cum KREV']]
y = df['BIT WEAR']
#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)
#creating model
lr = LinearRegression()
lr.fit(X_train, y_train)
lr_pred = lr.predict(X_test)

print('LR R2', metrics.r2_score(y_test, lr_pred))
print('LR MSE', metrics.mean_squared_error(y_test, lr_pred))

rr = Ridge(alpha=0.1, max_iter = 5000, solver = 'auto')
rr.fit(X_train, y_train)
rr_pred = rr.predict(X_test)

#showing results
print('RR R2', metrics.r2_score(y_test, rr_pred))
print('RR RMSE', sqrt(metrics.mean_squared_error(y_test, rr_pred))
print('RR MAE', metrics.mean_absolute_error(y_test, rr_pred

rr100 = Ridge(alpha=100, max_iter = 5000, solver = 'auto')
#  comparison with alpha value
rr100.fit(X_train, y_train)
rr100_pred = rr100.predict(X_test)

print('RR100 R2', metrics.r2_score(y_test, rr_pred))
print('RR100 RMSE', sqrt(metrics.mean_squared_error(y_test, rr_pred))
print('RR100 MAE', metrics.mean_absolute_error(y_test, rr_pred
```

Appendix 18. The Lasso Regressor (Python code)

```python
#creating the dataset
X = df[['ROP', 'MSE', 'WOB', 'TORQUE','RPM', 'FLOW','DEPTH OF CUT','BA', 'NOZZLE_VEL', 'JET_IMPACT', 'CROSS_FLOW', 'TE', 'cum TE', 'KREV', 'cum KREV']]
y = df['BIT WEAR']
#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)

#creating model
lasso = linear_model.Lasso(alpha=0.0001, copy_X=True, fit_intercept=True, max_iter=1000,
   normalize=False, positive=False, precompute=False, random_state=6,
   selection='random', tol=0, warm_start=True)
lasso.fit(X_train, y_train)
y_pred = lasso.predict(X_test)

# showing results
print('Lasso MAE', metrics.mean_absolute_error(y_test, y_pred))
print('Lasso RMSE', sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('Lasso MAPE', mean_absolute_percentage_error(y_test, y_pred))
print('Lasso R2', metrics.r2_score(y_test, y_pred))
```

Appendix 19. The Elastic Net Regressor (Python code)

```python
#creating the dataset
X = df[['ROP', 'MSE', 'WOB', 'TORQUE','RPM', 'FLOW','DEPTH OF CUT','BA', 'NOZZLE_VEL', 'JET_IMPACT', 'CROSS_FLOW', 'TE', 'cum TE', 'KREV', 'cum KREV']]
y = df['BIT WEAR']

#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)

#creating the model
regr = ElasticNet(alpha = 0.0001, l1_ratio = 0.1, tol = 0, warm_start = True, random_state=10)
regr.fit(X_train, y_train)
y_pred = regr.predict(X_test)

#showing results
print('Elastic R2', metrics.r2_score(y_test, y_pred))
print('Elastic RMSE', sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('Elastic MAE', metrics.mean_absolute_error(y_test, y_pred))
```

Appendix 20. The Decision Tree Regressor (Python code)

```python
#creating the dataset
X = df[['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'FLOW','DEPTH OF CUT','BA', 'NOZZLE_VEL', 'JET_IMPACT', 'CROSS_FLOW', 'TE', 'cum TE', 'KREV', 'cum KREV']]
y = df['BIT WEAR']

#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,
 random_state = 10)
#creating the model
model = DecisionTreeRegressor(criterion='friedman_mse', max_depth=14,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=2, min_samples_split=4,
           min_weight_fraction_leaf=0.0, presort=False, random_state=42,
           splitter='best')
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

#showing results
print('Tree R2', metrics.r2_score(y_test, y_pred))
print('Tree RMSE, sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('Tree MAE', metrics.mean_absolute_error(y_test, y_pred))
```

Appendix 21. The Random Forest Regressor (Python code)

```python
#creating the dataset
X = df[['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'FLOW','DEPTH OF CUT','BA', 'NOZZLE_VEL', 'JET_IMPACT', 'CROSS_FLOW', 'TE', 'cum TE', 'KREV', 'cum KREV']]
y = df['BIT WEAR']
#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)

model = RandomForestRegressor(max_features=5, max_depth = 13, min_samples_split=4, n_estimators=100, min_samples_leaf=2)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

#showing results
print('RForest R2', metrics.r2_score(y_test, y_pred))
print('RForest RMSE, sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('RForest MAE', metrics.mean_absolute_error(y_test, y_pred))
```

Appendix 22. The Ada Boost Regressor (Python code)

```python
#creating the dataset
X = df[['ROP', 'MSE', 'WOB', 'TORQUE', 'RPM', 'FLOW','DEPTH OF CUT','BA', 'NOZZLE_VEL', 'JET_IMPACT', 'CROSS_FLOW', 'TE', 'cum TE', 'KREV', 'cum KREV']]
y = df['BIT WEAR']
#creating the train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)

#creating the model
base_estimator = DecisionTreeRegressor(criterion='friedman_mse', max_depth=14,

           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=2, min_samples_split=4,
           min_weight_fraction_leaf=0.0, presort=False, random_state=42,
           splitter='best')

model = AdaBoostRegressor(base_estimator = base_estimator, random_state = 42, n_estimators=50, loss = 'square')
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

#showing results
print('AdaBoost R2', metrics.r2_score(y_test, y_pred))
print('AdaBoost RMSE, sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('AdaBoost MAE', metrics.mean_absolute_error(y_test, y_pred))
```