Predicting Bottomhole Temperatures : A Machine Learning Approach

Azharmadani Syed, Bala Srimani Durga Devi Chikkala, Guna Siddabathuni, Likitha Reddy kesara, Sandeep kokkula, Suguna Chandana Sibbena

ABSTRACT

Bottom-hole temperature (BHT) holds pivotal significance in the realm of petroleum engineering, serving as a critical parameter for well performance evaluation and reservoir characterization. Precise estimation of BHT enables well-informed decision-making across multiple operational contexts, including production optimisation, reservoir management, and drilling operations. This research uses a large dataset from the Norwegian continental shelf's Volve field to conduct a thorough analysis aimed at clarifying the complex dynamics of BHT and its consequences.

This project aims to provide practical insights for real-time decision making, predictive maintenance, and well optimisation in petroleum engineering through systematic exploration and predictive modelling. The study is divided into four main sections, including feature selection and machine learning model building for predictive analytics, visualisation of important metrics, data imputation techniques, and initial dataset overview and modification.

This paper integrates theoretical frameworks with empirical data to promote understanding and application in the field of BHT prediction techniques, building upon the key literature in the area. Novel methods like hybrid neural network optimisation provide valuable insights that could lead to improvements in BHT estimate efficiency and accuracy. By providing stakeholders navigating the challenges of hydrocarbon exploration and production with useful solutions and strategic insights, our research adds to the continuing conversation in petroleum engineering.

INTRODUCTION

Data has become a vital resource in the age of digitalization and technological growth, influencing important choices in many different fields and industries. The subject of petroleum engineering is one instance where the use of data analytics has grown. The disclosure by its licence partners of the complete dataset from the Norwegian continental shelf's Volve field has made a substantial contribution to the field's study and development.

This report's goal is to evaluate this large dataset, break down its various components, and develop prediction models to help in well optimisation, predictive maintenance, and real-time decision making. The dataset contains information on a number of characteristics, such as the number of hours spent in the field, the average downhole pressure and temperature, the average annular pressure, the average choke size percentage, the average wellhead pressure and temperature, the volume of oil and water, the volume of gas produced by the well, the volume of water injected, and the type of flow and well.

The analysis of this dataset has been systematically divided into four primary sections: the first is an overview and manipulation of the dataset; the second is an illustration of the dataset's different features; the third is the application of data imputation techniques; and the fourth is feature selection and the creation of a machine learning model for predictive analytics. Each component comprises a variety of sub-operations that have been carried out in order to analyse the data and extract insightful information.

LITERATURE REVIEW

Through the course of the project, we looked at several articles about BHT and BHP forecasts. These articles offered insightful analysis and useful techniques that shaped our strategy for BHT and BHP prediction.

A novel model for forecasting BHT during drilling operations is presented in the first article, "A New Model for Predicting Bottomhole Temperatures During Drilling Operations," by Al-Khdheeawi et al. (2020). The model takes into account a number of variables, including depth, circulation rate, and drilling fluid parameters. It is based on real-time data. The model developed by the authors was shown to be more accurate than previous models after being tested on multiple wells.

The second article, ”Spatial Prediction for Bottom Hole Temperature and Geothermal Gradient in Colombia” by Matiz-Le´on, provides a comprehensive study on the spatial prediction of BHT and geothermal gradient in Colombia. In the study, probabilistic techniques like Kriging and Sequential Gaussian Simulation (SGS) were used with deterministic techniques like minimal curvature and IDW. According to the study, when Kriging and SGS are used for spatial prediction, there are swings in the relative variance.

The third article, ”Effect of temperature variation on the accurate prediction of bottom-hole pressure in well drilling” by Ebrahim Hajidavalloo, Alireza Daneh-Dezfuli, and Ali Falavand Jozaei, focuses on the impact of various parameters on drilling fluid, which is a crucial part of a successful and safe drilling operation. The Bingham-plastic model is used in the study to develop the temperature and pressure distribution across the wellbore and to simulate drilling activities. For an oil-based mud, the study investigates the impacts of varying the inlet mud temperature, drilling fluid specific heat, and drilling fluid flow rate. As internal heat sources in the wellbore, it also takes into account the impacts of frictional pressure loss, pressure loss in the drill bit, and drill string rotation. The findings show that reliable bottom-hole pressure prediction requires thermal interaction. Changes in various parameters affect the wellbore's temperature distribution, which in turn affects the bottom-hole pressure. According to the study, bottom-hole pressure prediction may vary by as much as 685 psi, or around 2.30%, when taking into account variations in rheological parameters. To validate the model, field data was compared with its predictions.

The fourth article, titled ”Bottom Hole Pressure Estimation using Hybridization Neural Networks and Grey Wolves Optimization,” Amar, Zeraibi, and Redouane address the significance of bottom hole pressure (BHP) in production and reservoir studies. BHP plays a crucial role in well efficiency, completion design, and overall well strategies. However, there are drawbacks to using pressure gauges and other conventional methods for calculating BHP, including their high cost and difficulty processing noisy data. The authors provide a novel method for measuring BHP in vertical wells with multi-phase flow in order to get over these restrictions. They use an Artificial Neural Network (ANN) with a hidden layer made up of 12 neurons that were chosen by trial and error. This ANN is based on back-propagation training (BP-ANN). In addition, they use Grey Wolves optimisation to adjust the neural network's thresholds and weights. The researchers gathered 100 field data points from Algerian fields, including a wide variety of factors, in order to assess the correctness of their model. They evaluated how well their suggested model performed against two popular optimisation methods that are based on natural phenomena: Particle Swarm Optimisation (PSO) and Genetic Algorithm (GA). As the accompanying table shows, the results showed that the suggested model was more accurate than previous research when compared to other studies. By providing a more precise and economical approach, this research advances BHP estimate by overcoming the drawbacks of conventional techniques and improving comprehension and well operation optimisation across a range of field situations.

BUSINESS/ANALYTICS PROBLEM AND QUESTION FRAMING

The following business/analytics challenges and question formulations center on bottom hole temperature (BHT):

1.Predictive Maintenance Optimization:

Problem: What is the best way to use predictive analytics to maximize operational efficiency and minimize downtime by optimizing maintenance schedules based on bottom hole temperature fluctuations?

Question: Is it possible to create a machine learning model using past BHT data that forecasts equipment failures and provides proactive maintenance plans to avert possible problems?

2. Reservoir Performance Forecasting:

Problem: What factors determine bottom hole temperature changes inside the reservoir, and how can this information be used to accurately predict reservoir performance?

Question: How do changes in reservoir characteristics and fluid dynamics influence BHT trends? Can we develop prediction models for future BHT based on reservoir conditions and production strategies?

3. Increasing Operational Efficiency:

Problem: How can operational efficiency be increased by analyzing bottom hole temperature data and optimizing drilling and production activities accordingly?

Question: Are there any relationships between BHT fluctuations and operational parameters such flow rates, drilling fluid characteristics, and well depth? How can this information be used to improve operational strategies and resource utilization?

4. Risk Mitigation and Safety Enhancement

Problem: How can bottom hole temperature data be used to reduce operational risks and improve safety during drilling and production operations?

Question: How do aberrant BHT readings affect well integrity and safety? Can predictive analytics be used to identify potential safety risks and avoid accidents?

5. Environmental Impact Assessment

Problem: How do changes in bottom hole temperature affect environmental sustainability in oil and gas production, and what steps can be done to reduce environmental impact?

Question: How do variations in BHT impact the effectiveness of drilling fluid circulation and heat transfer in the reservoir? Can we create models to estimate the environmental impact of BHT fluctuations and then optimize drilling operations accordingly?

These questions aim to cover many aspects of petroleum engineering operations, with a particular emphasis on bottom hole temperature as a critical metric. Businesses in the oil and gas industry can optimise their operations, increase safety, and reduce environmental impact by analysing BHT data and using predictive analytics.

OBJECTIVE

The major goal of our project is to use modern data analytics techniques to analyse bottom hole temperature (BHT) data from the Volve field's extensive dataset on the Norwegian continental shelf. By focusing on BHT, we hope to acquire useful insights into reservoir behaviour, drilling operations, and production performance in the petroleum engineering area. Our goal is to investigate various elements of BHT variations, identify affecting factors, and develop prediction models to help with real-time decision making, predictive maintenance, and well optimisation operations. This analysis aims to optimise operational efficiency, improve safety measures, and reduce environmental effect in oil and gas production activities. Furthermore, our goal include using shared knowledge, technology, and novel data analysis methodologies to contribute to the advancement and sustainability of the energy business.

IMPACT AND VALUE

Analyzing bottom hole temperature (BHT) data from the Volve field dataset produces numerous significant results. First, by understanding the patterns and trends in BHT fluctuations, operators may make better judgements about drilling operations, reservoir management, and production optimization. This improves operating efficiency, reduces downtime, and increases output. Furthermore, predictive maintenance procedures built using BHT insights enable proactive interventions, reducing equipment failures and increasing overall operational reliability. Additionally, using BHT data improves safety measures by allowing for the anticipation and mitigation of possible hazards, resulting in a safer working environment for staff. Furthermore, BHT analysis aids in environmental impact assessment by evaluating the effectiveness of drilling operations and their consequences for ecosystem integrity, encouraging sustainability in oil and gas production activities.

Analyzing BHT data provides value in a variety of ways. To begin, improved operational efficiency saves oil and gas businesses money by reducing downtime and increasing output. Second, preventive maintenance practices lower maintenance costs and increase equipment lifespan, resulting in additional cost savings. Additionally, improved safety measures lead to fewer accidents and incidents, lowering associated costs and liabilities. Furthermore, by reducing environmental impact through educated decision-making based on BHT analysis, businesses demonstrate their commitment to sustainability, which can boost their reputation and stakeholder trust. Overall, using BHT data improves operational performance and efficiency while also increasing safety, lowering costs, and promoting environmental responsibility, providing significant value for the organization and the industry as a whole.

DATA

BACKGROUND OF THE DATASET

The information offered by the Volve licence partners includes a comprehensive collection of subsurface and operational data from the Volve field on the Norwegian continental shelf. This release is an important milestone because it contains the most extensive NCS data disclosure to date. Motivated by a desire to assist future energy innovators, the decision to share this dataset was made in June 2018, indicating a commitment to assisting research and study endeavours in the petroleum engineering sector.

Attributes of the Dataset

1. DATEPRD (Date of Record): Records the date of the data entry.

2. ON_STREAM_HRS (On Stream Hours): Measures the duration of operational activity in hours.

3. AVG_DOWNHOLE_PRESSURE (Average Downhole Pressure): Indicates the average pressure exerted at the downhole level in bars.

4.AVG_DOWNHOLE_TEMPERATURE (Average Downhole Temperature): Represents the average temperature recorded at the downhole level in degrees Celsius.

5. AVG_DP_TUBING (Average Differential Pressure of Tubing): Reflects the average pressure difference within the tubing in bars.

6. AVG_ANNULUS_PRESS (Average Annular Pressure):*Represents the average pressure within the annular space in bars.

7. AVG_CHOKE_SIZE_P (Average Choke Size Percentage): Measures the average percentage of choke size utilized.

8. AVG_WHP_P (Average Wellhead Pressure): Indicates the average pressure measured at the wellhead in bars.

9. AVG_WHT_P (Average Wellhead Temperature): Represents the average temperature measured at the wellhead in degrees Celsius.

10. BORE_OIL_VOL (Oil Volume from Well): Quantifies the volume of oil extracted from the well in cubic meters.

11. BORE_WAT_VOL (Water Volume from Well): Measures the volume of water extracted from the well in cubic meters.

12. BORE_GAS_VOL (Gas Volume from Well): Quantifies the volume of gas extracted from the well in cubic meters.

13. BORE_WI_VOL (Water Volume Injected): Indicates the volume of water injected into the well in cubic meters.

14. FLOW_KIND (Type of Flow): Represents the type of flow observed.

15. WELL_TYPE (Type of Well): Indicates the type of well under consideration.

Utility for Predicting BHT

The dataset contains a wealth of information that is particularly relevant to estimating bottom hole temperature (BHT). Variables such as average downhole pressure, average downhole temperature, and average wellhead temperature provide useful information on the reservoir's thermal behaviour. Predictive models for temperature changes within the wellbore can be built by examining the correlations between these factors and BHT. Furthermore, variables such as flow type, well type, and operational parameters help to understand the dynamic aspects impacting BHT. As a result, this dataset is an important resource for conducting in-depth analysis and developing prediction models to reliably forecast BHT.

QUALITY OF THE DATASET

The dataset produced by the Volve field has a large quantity of data, with 15,634 items and 24 columns. Upon analysis, certain columns have missing values, indicating potential data quality issues. For example, characteristics like ON_STREAM_HRS, AVG_DOWNHOLE_PRESSURE, AVG_DOWNHOLE_TEMPERATURE, and others have varied degrees of missingness, indicating possible gaps in the data gathering process or measurement inaccuracies. Furthermore, the existence of null values in specific columns may impair the reliability and correctness of future analysis.

Furthermore, while most columns contain numerical data types (such as float64 and int64), there are certain qualitative columns expressed as objects. This mixed data type composition demands careful treatment throughout data processing and analysis to ensure consistency and correctness.

In terms of statistical summary, the dataset gives information about the distribution and central tendency of many attributes. For instance, the mean and standard deviation of characteristics such as AVG_DOWNHOLE_PRESSURE and AVG_WHT_P provide preliminary insights into the operational circumstances inside the Volve field.

To ensure the integrity and reproducibility of our studies, we will address missing values, outliers, and inconsistencies in the dataset. Furthermore, extraneous columns such as _NPD_WELL_BORE_NAME_, _NPD_FIELD_CODE_, and _NPD_FACILITY_NAME_ will be removed to speed up further computations and analyses, improving the dataset's overall quality and usability for predictive modelling and decision-making.

Data Preprocessing:

Data Manipulation:

Step 1: By considering the columns in our dataset, the following columns will be eliminated as they are discovered to be unnecessary in the future computations:

- _NPD_WELL_BORE_NAME_
- _NPD_FIELD_CODE_
- _NPD_FACILITY_NAME_

Step 2: To begin with our research and computations, we must first organize our dataset set in a manageable manner. So, for achieving this we are doing the

1. **Identifying and Filtering the Queries:** As stated in the Introduction section, our goal will be to construct a model that can predict the bottomhole temperature of a production well. As the name says,

the ideal flow type or well type is production. As a result, wells that are currently producing will be screened and picked.

Screenshot of the well names that are in the production state:

```
Out[12]: array(['15/9-F-1 C', '15/9-F-11', '15/9-F-12', '15/9-F-14', '15/9-F-15 D',
                '15/9-F-5'], dtype=object)
```

2. **Datetime Manipulations:** In this step the observations will be sorted in ascending order by date to ensures that the data is arranged chronologically from the earliest to the latest date.
3. **Adding new columns (years):**

This section divides the **DATEPRD** column, which holds datetime values, into entire date and year-only columns. This will make plotting more convenient in following stages.

Out[17]:

	Years	DATEPRD	WELL_BORE_CODE	NPD_WELL_BORE_CODE	NPD_WELL_BORE_NAME	NPD_FIELD_CODE	NPD_FIELD_NAME	NPD_FACILITY_CODE
4967	2008	2008-02-12	NO 15/9-F-14 H	5351	15/9-F-14	3420717	VOLVE	369304
1911	2008	2008-02-12	NO 15/9-F-12 H	5599	15/9-F-12	3420717	VOLVE	369304
4968	2008	2008-02-13	NO 15/9-F-14 H	5351	15/9-F-14	3420717	VOLVE	369304
1912	2008	2008-02-13	NO 15/9-F-12 H	5599	15/9-F-12	3420717	VOLVE	369304
1913	2008	2008-02-14	NO 15/9-F-12 H	5599	15/9-F-12	3420717	VOLVE	369304

5 rows × 25 columns

Finally, the cumulative production values for each phase will be calculated and identified as follows:

_BORE_OIL_CUM_, represents cumulative oil production

_BORE_GAS_CUM_, represents cumulative gas productions

_BORE_WAT_CUM_, represents cumulative water production

CUM columns of BORE_OIL, GAS and WAT

```
]: df_production.loc[:, 'BORE_OIL_CUM'] = df_production.loc[:, 'BORE_OIL_VOL'].cumsum()
   df_production.loc[:, 'BORE_GAS_CUM'] = df_production.loc[:, 'BORE_GAS_VOL'].cumsum()
   df_production.loc[:, 'BORE_WAT_CUM'] = df_production.loc[:, 'BORE_WAT_VOL'].cumsum()
```

```
]: df_production['BORE_OIL_CUM']
```

```
]: 4967           0.00
   1911         284.65
   4968         284.65
   1912        2154.35
   1913        5278.44
               ...
   4966     10036900.57
   8022     10036900.57
   1910     10037080.61
   9000     10037080.61
   15632    10037080.61
   Name: BORE_OIL_CUM, Length: 9161, dtype: float64
```

The cumulative production values (BORE_OIL_CUM, BORE_GAS_CUM, and BORE_WAT_CUM) show the entire amount of each phase (oil, gas, and water) produced up to a specific time point. These values are obtained by adding the respective phase production values across time. Which will be uses in **Performance Tracking, Reserve Estimation, Revenue Calculation.**

Understanding the data:

Step 1: <u>Total production volumes of oil, gas, and water:</u>

- Using the DataFrame df_production, we calculate the total volume of oil, gas, and water.
- Total oil, gas, and water quantities are computed by adding the columns in the Data Frame.
- Total three-phase production volume is calculated by combining oil, gas, and water volumes.
- The output offers information on each fluid's production volume and total combined production.
- These insights help to measure production efficiency, identify trends, and influence decision-making in resource management and operational planning in the oil and gas industry.

Output:

```
Total oil produced is 10037080.61 cubic meters
Total oil produced is 1475370435.94 cubic meters
Total oil produced is 15318578.35 cubic meters
The total amount produced is 35392740 cubic meters, 222613635 barrels.
```

Benefit:

- Calculating overall production volumes offers valuable insights into the production profile, aiding in resource allocation, optimization, and forecasting.
- Monitoring production amounts over time allows for evaluating well or field performance and identifying trends.
- Understanding the distribution of fluids produced facilitates optimal resource management and operational planning.
- Informed decisions based on these insights can enhance production efficiency, resource utilization, and operational performance in the oil and gas industry.

Step 2: Describe

From the analysis what we understand is

Operational Metrics:

- Wells were on stream for an average of 20.17 hours, ranging from 0 to 25 hours.
- The average downhole pressure is at 181.80 units, with a standard deviation of roughly 109.71 units.
- The average downhole temperature is about 77.16 units, with a standard deviation of 45.66 units.

Production Volumes:

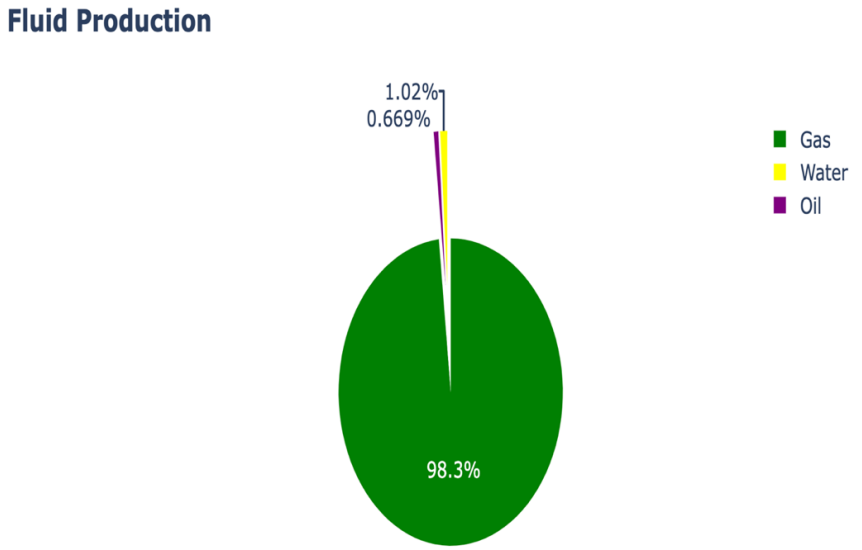
- The average volume of oil produced per bore is 1095.63 cubic meters, with a standard deviation of 1323.54.
- The average volume of gas generated per bore is about 161,049.06 cubic meters, with a standard deviation of 188,136.41 cubic meters.
- The average volume of water generated each bore is about 1672.15 cubic meters, with a standard deviation of 1706.98 cubic meters.
- The entire cumulative output (oil, gas, and water) ranges between 7.37 million and 1.50 billion cubic meters.

Choke and Tubing Information:

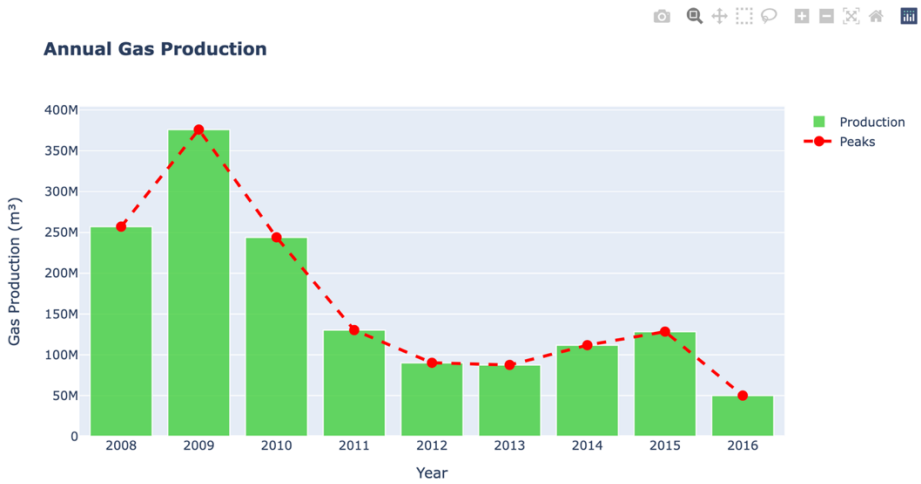
- The average choke size percentage is 55.17%, with a standard deviation of about 36.69%.
- The average tubing pressure is at 45.38 units, with a standard deviation of roughly 24.75 units.
- The average annulus pressure is around 67.73 units, with a standard deviation of roughly 27.72 units.

Step 3: **Visualizations of Production Data without Imputation:**

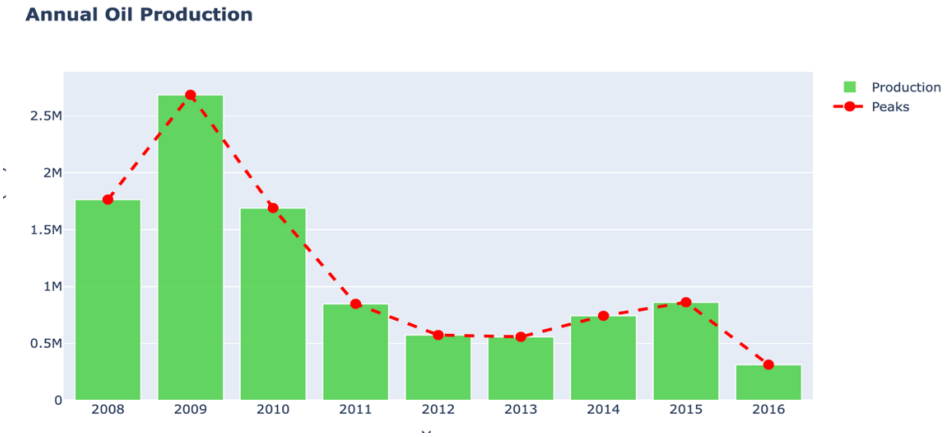
Generating a pie chart that visually represents the distribution of fluid production (Oil, Gas, Water) in the dataset, providing an intuitive way to understand the relative contributions of each phase **:**



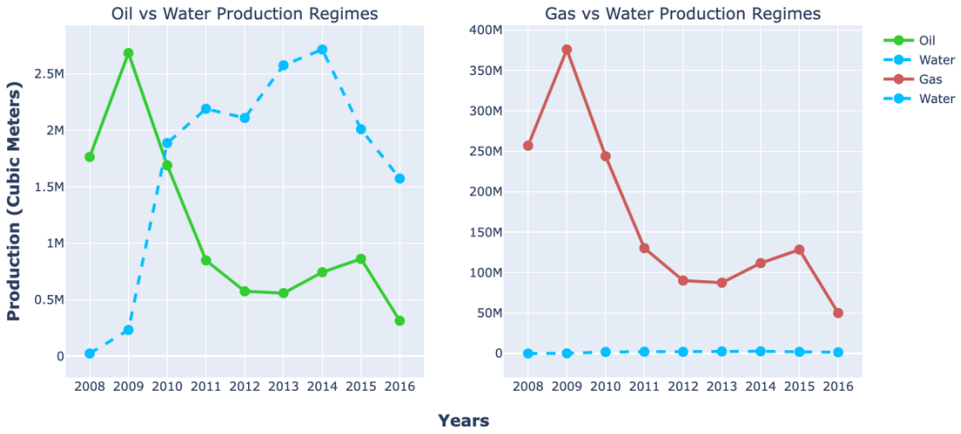
Generating a plot that visualizes the annual gas production over the years. The bar plot provides a clear representation of the gas production volume for each year, while the overlaid scatter plot helps identify any significant peaks or trends in the production data. This visualization aids in understanding the historical trends and patterns in gas production.



Generating a plot illustrating the annual oil production over the years. The bar plot provides a clear depiction of the oil production volume for each year, while the overlaid scatter plot helps identify any notable peaks or trends in the production data. This visualization assts in understanding the historical trends and patterns in oil production.



created a subplot with two panels to compare oil and gas production with water production over the years. Each panel represents different production regimes: oil vs. water in the first panel and gas vs. water in the second. The visualization helps understand the relationship and trends between oil, gas, and water production over time.



Observations from the above plots:

- Gas emerges as the main phase, with production volumes exceeding those of the other two phases.
- The reservoir has the features of a volatile oil reservoir supported by an aquifer.
- After early production, the reservoir quickly enters the saturated region.
- A probable enhanced oil recovery operation could be underway, as the oil production regime shows a minor increase after 2013.

Data Wrangling:

Step 1: Identifying Missingess:

Identified missing values by df_production.isnull().sum() code and we can see particularly in columns related to downhole pressure, temperature, tubing, annulus pressure, and choke size, with varying degrees of completeness.

The percentage of null values for each attribute in the dataset is as follows:

AVG_DOWNHOLE_PRESSURE: 1.98%

AVG_DOWNHOLE_TEMPERATURE: 1.98%

AVG_DP_TUBING: 1.98%

AVG_ANNULUS_PRESS: 13.87%

AVG_CHOKE_SIZE_P: 2.64%

AVG_WHP_P: 0.07%

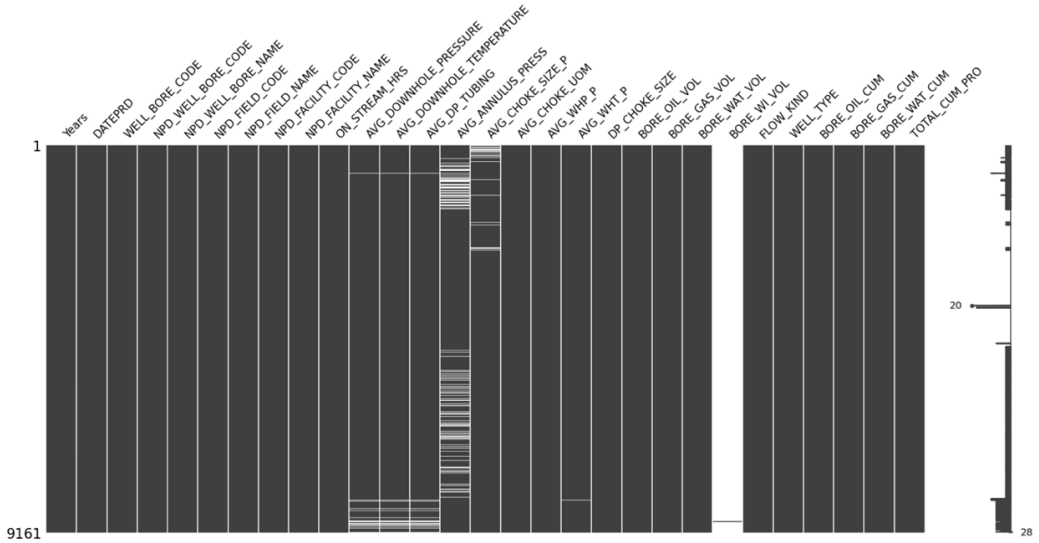
AVG_WHT_P: 0.16%

DP_CHOKE_SIZE: 0.07%

BORE_WI_VOL: 99.84%

Step 2: Data Imputation:

Missing value matrix of our dataset which is used to visualize the distribution of missing values in our DataFrame which will helps in understanding the pattern and extent of missingness in the dataset.



Step 3: Handling Missing Values

Upon examining the total count of missing values across all columns, it became evident that the column AVG_DOWNHOLE_TEMPERATURE, designated as our target variable, exhibited a significant number of missing entries. Consequently, a decision was made to eliminate rows containing missing values within this column.

```
In [40]: df.isnull().sum()

Out[40]: DATEPRD                        0
         WELL_BORE_CODE                 0
         NPD_WELL_BORE_CODE             0
         NPD_WELL_BORE_NAME             0
         NPD_FIELD_CODE                 0
         NPD_FIELD_NAME                 0
         NPD_FACILITY_CODE              0
         NPD_FACILITY_NAME              0
         ON_STREAM_HRS                285
         AVG_DOWNHOLE_PRESSURE       6654
         AVG_DOWNHOLE_TEMPERATURE    6654
         AVG_DP_TUBING               6654
         AVG_ANNULUS_PRESS           7744
         AVG_CHOKE_SIZE_P            6715
         AVG_CHOKE_UOM               6473
         AVG_WHP_P                   6479
         AVG_WHT_P                   6488
         DP_CHOKE_SIZE                294
         BORE_OIL_VOL                6473
         BORE_GAS_VOL                6473
         BORE_WAT_VOL                6473
         BORE_WI_VOL                 9928
         FLOW_KIND                      0
         WELL_TYPE                      0
         dtype: int64

In [41]: df = df.dropna(subset=['AVG_DOWNHOLE_TEMPERATURE'])
```

Following the removal of rows containing missing values, it became apparent that the column BORE_WI_VOL also harbors a notable amount of missing data. As a strategic measure, it was determined that the entire column would be dropped from the dataset.

```
In [42]: df.isnull().sum()

Out[42]: DATEPRD                        0
         WELL_BORE_CODE                 0
         NPD_WELL_BORE_CODE             0
         NPD_WELL_BORE_NAME             0
         NPD_FIELD_CODE                 0
         NPD_FIELD_NAME                 0
         NPD_FACILITY_CODE              0
         NPD_FACILITY_NAME              0
         ON_STREAM_HRS                  0
         AVG_DOWNHOLE_PRESSURE          0
         AVG_DOWNHOLE_TEMPERATURE       0
         AVG_DP_TUBING                  0
         AVG_ANNULUS_PRESS           1259
         AVG_CHOKE_SIZE_P             240
         AVG_CHOKE_UOM                  0
         AVG_WHP_P                      0
         AVG_WHT_P                      0
         DP_CHOKE_SIZE                  0
         BORE_OIL_VOL                   0
         BORE_GAS_VOL                   0
         BORE_WAT_VOL                   0
         BORE_WI_VOL                 8980
         FLOW_KIND                      0
         WELL_TYPE                      0
         dtype: int64

In [43]: df = df.drop(columns=['BORE_WI_VOL'])
```

Upon the removal of the column BORE_WI_VOL, it was observed that certain missing values persisted in the columns AVG_ANNULUS_PRESS and AVG_CHOKE_SIZE_P. To address these missing values, a decision was made to implement a simple imputer method.

```
df_production.isnull().sum()

Years                       0
DATEPRD                     0
WELL_BORE_CODE              0
NPD_WELL_BORE_CODE          0
NPD_WELL_BORE_NAME          0
NPD_FIELD_CODE              0
NPD_FIELD_NAME              0
NPD_FACILITY_CODE           0
NPD_FACILITY_NAME           0
ON_STREAM_HRS               0
AVG_DOWNHOLE_PRESSURE       0
AVG_DOWNHOLE_TEMPERATURE    0
AVG_DP_TUBING               0
AVG_ANNULUS_PRESS        1259
AVG_CHOKE_SIZE_P          240
AVG_CHOKE_UOM               0
AVG_WHP_P                   0
AVG_WHT_P                   0
DP_CHOKE_SIZE               0
BORE_OIL_VOL                0
BORE_GAS_VOL                0
BORE_WAT_VOL                0
FLOW_KIND                   0
WELL_TYPE                   0
BORE_OIL_CUM                0
BORE_GAS_CUM                0
BORE_WAT_CUM                0
TOTAL_CUM_PRO               0
dtype: int64
```

Upon the implementation of the simple imputer method, the following results were obtained.

SimpleImputer

```
In [51]: from sklearn.impute import SimpleImputer

In [52]: imputer = SimpleImputer(strategy='median')  # Replace 'mean' with 'median', 'most_frequent' or 'constant'
         df['AVG_ANNULUS_PRESS'] = imputer.fit_transform(df[['AVG_ANNULUS_PRESS']])

In [53]: imputer = SimpleImputer(strategy='median')  # Replace 'mean' with 'median', 'most_frequent' or 'constant'
         df['AVG_CHOKE_SIZE_P'] = imputer.fit_transform(df[['AVG_CHOKE_SIZE_P']])

In [54]: df.isnull().sum()

Out[54]: DATEPRD                    0
         WELL_BORE_CODE             0
         NPD_WELL_BORE_CODE         0
         NPD_WELL_BORE_NAME         0
         NPD_FIELD_CODE             0
         NPD_FIELD_NAME             0
         NPD_FACILITY_CODE          0
         NPD_FACILITY_NAME          0
         ON_STREAM_HRS              0
         AVG_DOWNHOLE_PRESSURE      0
         AVG_DOWNHOLE_TEMPERATURE   0
         AVG_DP_TUBING              0
         AVG_ANNULUS_PRESS          0
         AVG_CHOKE_SIZE_P           0
         AVG_CHOKE_UOM              0
         AVG_WHP_P                  0
         AVG_WHT_P                  0
         DP_CHOKE_SIZE              0
         BORE_OIL_VOL               0
         BORE_GAS_VOL               0
         BORE_WAT_VOL               0
         FLOW_KIND                  0
         WELL_TYPE                  0
         dtype: int64
```

Following the completion of missing value handling procedures, it was determined that the dataset no longer contains any missing values. The subsequent step involves the removal of unnecessary columns from the dataset to facilitate the prediction of the target variable.

Dropping Irrelevant columns

```
columns_to_drop = ['WELL_BORE_CODE', 'NPD_WELL_BORE_CODE', 'NPD_WELL_BORE_NAME',
       'NPD_FIELD_CODE', 'NPD_FIELD_NAME', 'NPD_FACILITY_CODE',
       'NPD_FACILITY_NAME','AVG_CHOKE_UOM']
```

```
df.drop(columns_to_drop, axis = 1, inplace = True)
```

```
df_cat = df.select_dtypes(include= 'O')
df_num = df.select_dtypes(include= np.number)
```

```
df.dtypes
```

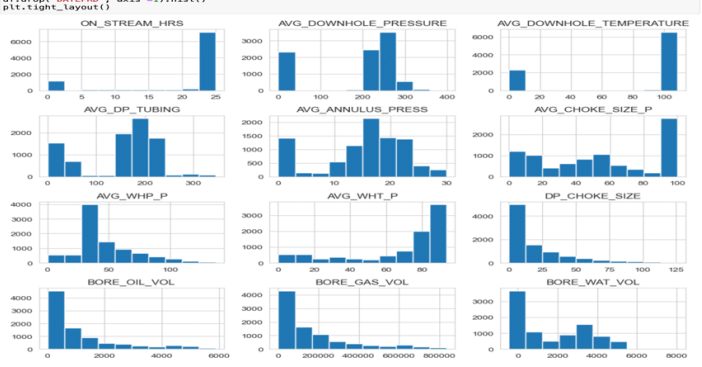
```
DATEPRD                    datetime64[ns]
ON_STREAM_HRS                     float64
AVG_DOWNHOLE_PRESSURE             float64
AVG_DOWNHOLE_TEMPERATURE          float64
AVG_DP_TUBING                     float64
AVG_ANNULUS_PRESS                 float64
AVG_CHOKE_SIZE_P                  float64
AVG_WHP_P                         float64
AVG_WHT_P                         float64
DP_CHOKE_SIZE                     float64
BORE_OIL_VOL                      float64
BORE_GAS_VOL                      float64
BORE_WAT_VOL                      float64
FLOW_KIND                          object
WELL_TYPE                          object
dtype: object
```

```
# All the datatypes are assigned perfectly.
```

EDA

Prior to delving into detailed statistical analysis, an initial assessment of the frequency distributions for each variable within the dataset was conducted. This overview is crucial for identifying underlying data structures and informing subsequent analyses. Below is a summary of the distributions observed for each key operational parameter.

```
In [81]: df.drop('DATEPRD', axis =1).hist()
         plt.tight_layout()
```



Based on the histograms and their distribution, here's an analysis of each variable:

ON_STREAM_HRS: The distribution is skewed to the right, with most values clustered near the lower end, suggesting that fewer operational hours are more common.

AVG_DOWNHOLE_PRESSURE: This distribution has two peaks, suggesting that there are two common values for average downhole pressure where the data clusters, which could indicate two different operational states or conditions in the well.

AVG_DOWNHOLE_TEMPERATURE: The distribution is heavily skewed to the right, with most wells having a low average downhole temperature and a few wells with much higher temperatures.

AVG_DP_TUBING: The data shows a right skew, indicating that higher pressure differentials are less common.

AVG_ANNULUS_PRESS: The distribution is skewed right, with a concentration of values towards the lower end of the scale.

AVG_CHOKE_SIZE_P: This histogram has a multi-modal distribution, suggesting different choke sizes are used depending on the specific operational requirements or strategies.

AVG_WHP_P: The distribution here is right-skewed, indicating that lower wellhead pressures are more common.

AVG_WHT_P: The histogram shows a right-skewed distribution, with most data points indicating lower wellhead temperatures and fewer instances of higher temperatures.

DP_CHOKE_SIZE: This variable has a right-skewed distribution as well, indicating that smaller choke sizes, and therefore greater choke-induced pressure differentials, are more common.

BORE_OIL_VOL: The oil volume produced shows a very strong right skew, indicating that most of the time, the production volume is on the lower end, with few instances of very high production.

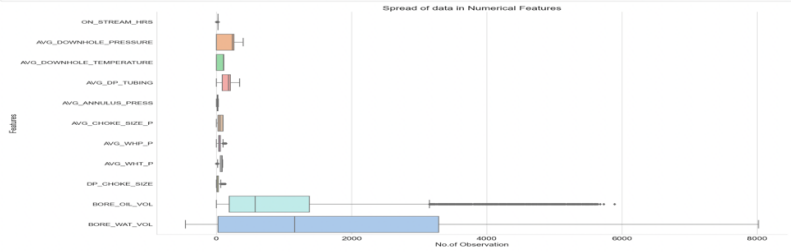
BORE_GAS_VOL: The gas volume produced is also right-skewed, but with a much wider spread, suggesting that while low to moderate production volumes are common, there are also instances of very high gas production.

BORE_WAT_VOL: Water volume shows a right skew similar to oil and gas, indicating lower volumes are more common, with a few higher volume cases.

The skewness in these histograms suggests that for many of the measured variables, lower values are more common, with fewer instances of high values. This is typical in operational data for oil and gas production where a majority of the time, production is steady or controlled at lower values, and only occasionally there are surges or spikes in production or pressure/temperature readings. The multi-modal distributions, such as seen in the average downhole pressure and choke size, might indicate the presence of distinct groups or operational modes within the dataset.

Box Plot:

```
In [72]: sns.set_style("whitegrid")
         plt.subplots(figsize=(21, 14))
         color = sns.color_palette('pastel')
         sns.boxplot(data = df_num.drop('BORE_GAS_VOL',axis = 1), orient='h', palette=color)
         plt.title('Spread of data in Numerical Features', size = 20)
         plt.xlabel('No.of Observation', size = 17)
         plt.ylabel('Features', size = 17)
         plt.xticks(size = 17)
         plt.yticks(size = 15)
         sns.despine()
         plt.show()
```



Boxplot Distribution Analysis

The accompanying boxplot figure provides a comprehensive overview of the spread of numerical data across several operational features within our dataset. The orientation of the boxplots is horizontal, facilitating a clearer comparison across the variables. The variables examined include operational hours, downhole pressure and temperature, differential pressure across tubing, annulus pressure, choke size, wellhead pressure and temperature, and choke size pressure differential, alongside the volume of oil, gas, and water production.

Key observations from the boxplots are as follows:

Operational Hours (ON_STREAM_HRS): Exhibits a relatively tight interquartile range (IQR), with a median value indicating consistent operational behavior across the dataset.

(AVG_DOWNHOLE_PRESSURE) & (AVG_DOWNHOLE_TEMPERATURE): These parameters show a varied range with outliers suggesting occasional deviations from typical operational conditions.

Differential Pressure Across Tubing (AVG_DP_TUBING): Displays a wider IQR, which indicates variability in this measurement across different operations.

Annulus Pressure (AVG_ANNULUS_PRESS): Has a compact box but with outliers that may correspond to uncommon but significant operational scenarios.

(AVG_CHOKE_SIZE_P) and (AVG_WHP_P): These features present with a median close to the lower quartile, suggesting a skew towards lower values.

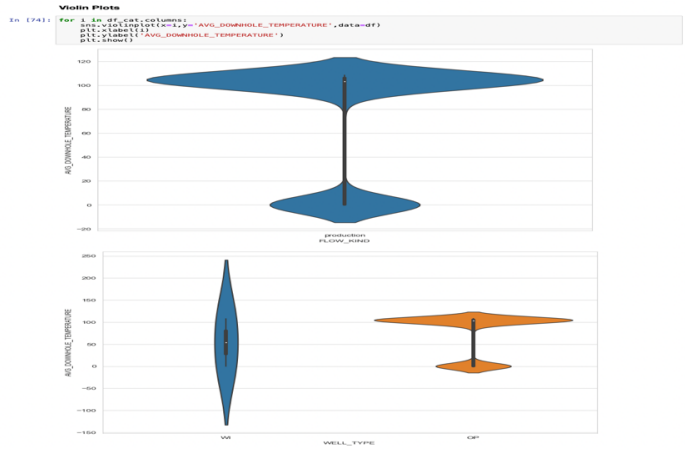
Wellhead Temperature (AVG_WHT_P): The boxplot indicates a narrow range, hinting at consistent temperature control in operational procedures.

Choke Size Pressure Differential (DP_CHOKE_SIZE): Reveals outliers which could represent moments of high pressure differential, important for understanding flow conditions.

Oil (BORE_OIL_VOL) and Water Volume (BORE_WAT_VOL): These boxplots are notably elongated with several outliers, especially for oil, indicating a significant variation in production volumes, which could be attributed to well productivity or operational interventions.

The color palette chosen ('pastel') ensures clarity and visual distinction between each feature's distribution. This visual analysis is a prelude to identifying patterns, anomalies, and trends in the dataset, serving as a foundation for further statistical inquiry and operational optimization.

Violin Plot



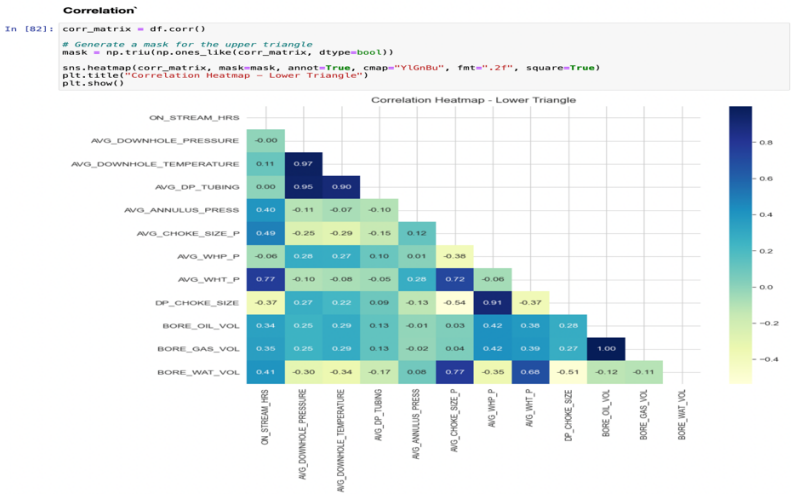
Violin Plot Analysis of Average Downhole Temperature

The provided violin plots afford a detailed visualization of the distribution of average downhole temperature stratified by flow kind and well type. In the first plot, the variable 'FLOW_KIND' is represented with only one category, 'production,' depicting a wide distribution of temperatures with most measurements clustering around 60 degrees. This spread suggests a high degree of variability in temperature conditions for production flows.

In the second plot, the average downhole temperatures are segmented by 'WELL_TYPE' into two categories, denoted 'WI' and 'OP'. The 'WI' well type is characterized by a distribution with a narrow focus around the median value, indicative of a more consistent thermal profile within these wells. Conversely, the 'OP' well type exhibits a broader temperature distribution, implying a wider range of temperature conditions and, thus, potentially more diverse operational characteristics or geological conditions.

These violin plots are instrumental in revealing the underlying temperature distributions for different categories, providing an invaluable perspective on the thermal behavior associated with different well operations. Such visual analysis is critical for the identification of patterns that may influence operational strategies and decision-making processes in the management of well sites.

Correlation



Correlation Heatmap Visualization

The correlation heatmap provides a succinct visual summary of the pairwise correlations between the operational parameters within our dataset. By employing a lower triangle format, redundancies are eliminated, allowing for a focused interpretation of the relationships.

Key observations include:

A strong positive correlation is evident between 'AVG_DOWNHOLE_PRESSURE' and 'AVG_DOWNHOLE_TEMPERATURE,' suggesting that as the pressure within a well increases, the temperature tends to rise as well.

'AVG_WHT_P' (average wellhead temperature) displays notable positive correlations with 'AVG_DOWNHOLE_PRESSURE' and 'AVG_DOWNHOLE_TEMPERATURE,' indicating a relationship between downhole conditions and surface measurements.

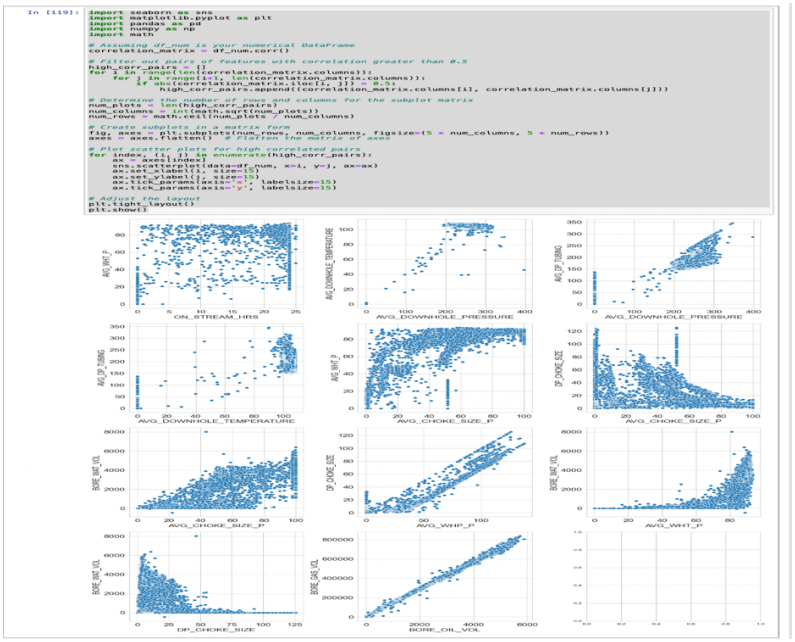
'DP_CHOKE_SIZE' exhibits a strong positive correlation with 'AVG_CHOKE_SIZE_P,' which could be reflective of the direct impact of choke size adjustments on pressure differentials.

'BORE_GAS_VOL' (volume of gas produced) shows a perfect correlation with itself, as expected, and positive correlations with 'BORE_OIL_VOL' and 'BORE_WAT_VOL,' highlighting the interdependence of hydrocarbon and water production volumes.

It is essential to note that while correlations provide insights into potential relationships, they do not imply causation. Furthermore, the heat map's use of a diverging color scheme effectively distinguishes between different magnitudes of correlation, facilitating the rapid identification of both strong and weak associations.

The heatmap functions as a preliminary tool in our exploratory data analysis, guiding further statistical testing and model building to ascertain the significance and nature of these observed correlations.

Scatter Plot



Scatter Plot Analysis for Highly Correlated Variable Pairs

In our comprehensive data examination, we identified pairs of variables with strong correlations (greater than 0.5) and visualized their relationships using scatter plots. These plots are invaluable for discerning the nature of the correlations, whether positive or negative, and for assessing the strength of these relationships.

Highlights from the scatter plot matrix include:

AVG_WHT_P vs. AVG_DOWNHOLE_PRESSURE: A clear positive correlation indicates that as downhole pressure increases, the temperature at the wellhead tends to increase as well.

AVG_DP_TUBING vs. AVG_DOWNHOLE_PRESSURE : This positive correlation suggests a relationship between the pressure within the wellbore and the differential pressure experienced across the tubing.

DP_CHOKE_SIZE vs. AVG_WHP_P: The strong positive correlation here may imply that adjustments in choke size can significantly affect wellhead pressure.

BORE_OIL_VOL vs. BORE_GAS_VOL: An extremely tight positive correlation reveals that oil and gas production volumes are closely linked, likely reflecting the co-production of these hydrocarbons.

Each plot has been generated with a focus on clarity and scale, ensuring that data points are distinguishable and axes are appropriately labeled for ease of interpretation. Notably, the consistent sizing of axes and text across the matrix allows for direct comparison between plots.

The scatter plot matrix serves as a critical tool in our exploratory data analysis, revealing insights that drive further investigations into the underlying physical phenomena or operational processes. These visual representations will underpin our subsequent modeling efforts and can inform more nuanced strategies for production optimization and risk management.

Statistical Test for Numerical Columns:

Performed the Pearson correlation coefficients and p- values between numeric variables and AVG_DOWNHOLE_TEMPERATURE to assess their linear relationship. Which will help in identifying the variables potentially influencing downhole temperature, aiding in understanding oil well behavior.

Now, let's interpret the output:

```
ON_STREAM_HRS: p-value = 7.86776611404082e-24
AVG_DOWNHOLE_PRESSURE: p-value = 0.0
AVG_DP_TUBING: p-value = 0.0
AVG_ANNULUS_PRESS: p-value = 4.016242372255136e-11
AVG_CHOKE_SIZE_P: p-value = 1.3791357935463643e-16
AVG_WHP_P: p-value = 1.126122575238344e-154
AVG_WHT_P: p-value = 3.154636842923036e-13
DP_CHOKE_SIZE: p-value = 1.15042942670228e-100
BORE_OIL_VOL: p-value = 2.6451794011716775e-173
BORE_GAS_VOL: p-value = 3.039089729766467e-170
BORE_WAT_VOL: p-value = 3.096040384134317e-247
```

For ON_STREAM_HRS: p-value = 7.86776611404082e-24
This indicates a very low p-value, suggesting a significant correlation between 'ON_STREAM_HRS' and 'AVG_DOWNHOLE_TEMPERATURE'.

All these variables have extremely low p-values (around 0 in scientific notation), showing strong evidence against the null hypothesis. As a result, all of these variables appear to be significantly correlated with 'AVG_DOWNHOLE_TEMPERATURE'.

In summary, the results show that the bulk of the quantitative variables in the dataset have statistically **significant correlations with the target variable**, 'AVG_DOWNHOLE_TEMPERATURE'.

Multiple linear regression analysis using the Ordinary Least Squares (OLS) method:
We used this method for investigating the association between independent variables (features) and a target variable ('AVG_DOWNHOLE_TEMPERATURE'). It adds a constant term to the independent variables, fits the regression model, and then outputs a model summary that includes coefficients, standard errors, t-statistics, p-values, and other important statistics. This approach aids in determining which independent variables have a substantial impact on the target variable, allowing us to better understand the causes controlling downhole temperature in oil wells.

Output:

OLS Regression Results

Dep. Variable:	AVG_DOWNHOLE_TEMPERATURE	R-squared:	0.972
Model:	OLS	Adj. R-squared:	0.972
Method:	Least Squares	F-statistic:	2.869e+04
Date:	Wed, 24 Apr 2024	Prob (F-statistic):	0.00
Time:	21:03:51	Log-Likelihood:	-30942.
No. Observations:	8980	AIC:	6.191e+04
Df Residuals:	8968	BIC:	6.199e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	8.0100	0.388	20.665	0.000	7.250	8.770
ON_STREAM_HRS	0.9157	0.018	49.672	0.000	0.880	0.952
AVG_DOWNHOLE_PRESSURE	0.4605	0.003	144.107	0.000	0.454	0.467
AVG_DP_TUBING	-0.1052	0.004	-24.461	0.000	-0.114	-0.097
AVG_ANNULUS_PRESS	-0.1824	0.012	-15.095	0.000	-0.206	-0.159
AVG_CHOKE_SIZE_P	-0.0750	0.004	-17.973	0.000	-0.083	-0.067
AVG_WHP_P	0.2959	0.013	23.437	0.000	0.271	0.321
AVG_WHT_P	-0.1582	0.008	-20.234	0.000	-0.173	-0.143
DP_CHOKE_SIZE	-0.5133	0.015	-34.981	0.000	-0.542	-0.485
BORE_OIL_VOL	-0.0011	0.001	-0.933	0.351	-0.004	0.001
BORE_GAS_VOL	1.107e-05	8.63e-06	1.283	0.200	-5.85e-06	2.8e-05
BORE_WAT_VOL	-0.0020	9.41e-05	-20.866	0.000	-0.002	-0.002

Omnibus:	2294.351	Durbin-Watson:	0.250
Prob(Omnibus):	0.000	Jarque-Bera (JB):	21678.737
Skew:	-0.953	Prob(JB):	0.00
Kurtosis:	10.369	Cond. No.	1.20e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.2e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

From above we can see that:

- The regression model has an excellent overall fit (R-squared = 0.972), indicating that the independent variables explain 97.2% of the variability in 'AVG_DOWNHOLE_TEMPERATURE'.

- Several variables, including 'ON_STREAM_HRS', 'AVG_DOWNHOLE_PRESSURE', 'AVG_DP_TUBING', 'AVG_CHOKE_SIZE_P', 'AVG_WHP_P', 'AVG_WHT_P', and 'DP_CHOKE_SIZE', have significant coefficients ($p < 0.05$), indicating a significant impact on downhole temperature.

- However, 'BORE_OIL_VOL' and 'BORE_GAS_VOL' exhibit non-significant coefficients ($p > 0.05$), indicating that they may not have a major impact on downhole temperature.

- The F-statistic (2.869e+04) with a low probability confirms the model's overall significance, showing its dependability.

- The Durbin-Watson value (0.250) indicates that there may be autocorrelation in the model residuals.

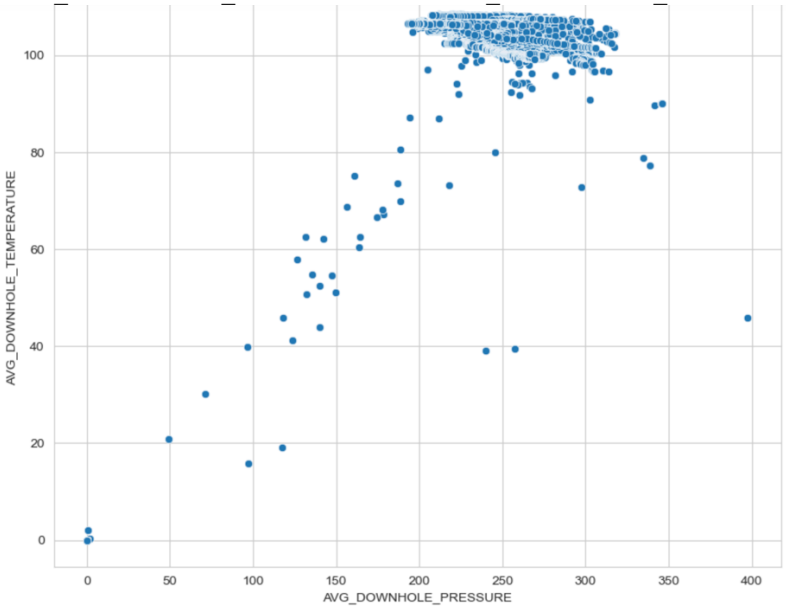
- The condition number (1.2e+06) suggests that the independent variables may be multicollinear, which could alter the coefficients' stability and interpretation.

Inference:
From the above statistical test, p-values lower than a selected level of significance (e.g., 0.05) usually indicate a significant association between the numerical variable and target variable. In our case almost all the p-values are less than 0.05, which means that most of the factors have significant association with the target variable Downholw Temperature at 0.05 level.

However, choosing significance level is arbitrary and could be adjusted according to our specific needs and circumstances of our analysis. Statistically, while significance is critical in evaluating whether or not a predictor is important; there are other things to consider as well.

Data Discrepancy Analysis:

Step 1: Initial Exploration: Exploratory data analysis (EDA) was performed using a scatter plot of 'AVG_DOWNHOLE_PRESSURE' and 'AVG_DOWNHOLE_TEMPERATURE'.



Inference:

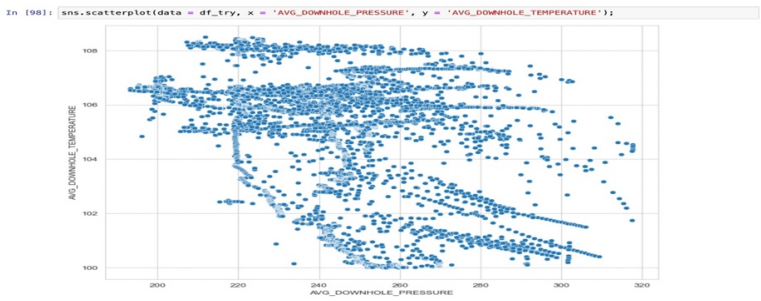
• As we can see from the above plot, there is a very positive correlation between the temperature and pressure.
• But infact there are many miscalculations in the data.
• There are many 0 values of the temperature in our dataset.
• So, we have removed the 0 values and the temperature would be greater than 100 at the bottom hole for sure, so let's consider all the values above 100 for the temperature and do the model building.

Step 2: Data Cleaning:

- Removed zero values from'AVG_DOWNHOLE_TEMPERATURE'.
- Filtered the dataset to only include temperatures above 100, as values below this level were likely incorrect.

Step 3: Revised Visualization:

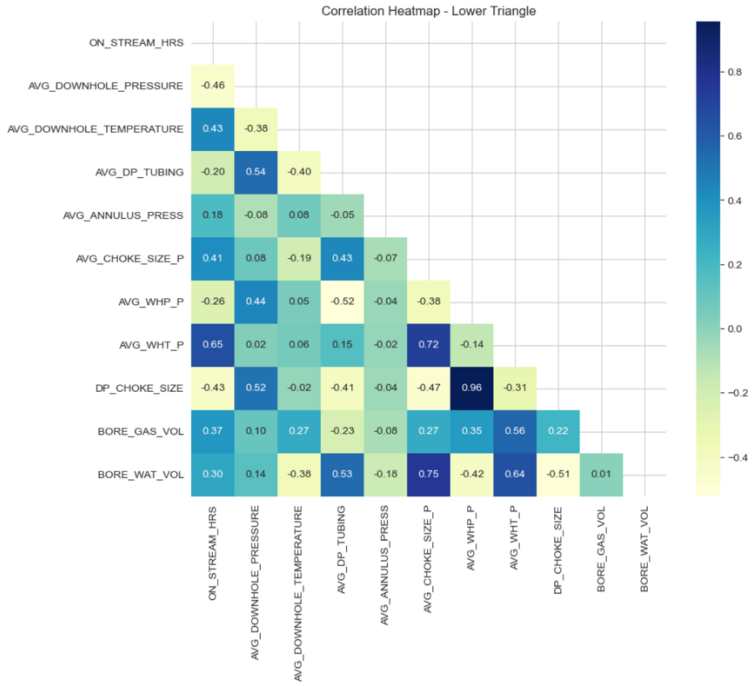
- To appropriately portray the data, we visualized the relationship between 'AVG_DOWNHOLE_PRESSURE' and 'AVG_DOWNHOLE_TEMPERATURE' after cleaning.
- There was a clearer correlation pattern between the variables.



Inference: We can see there is no perfect correlation from the above plot

Step 4: Correlation Reassessment:

- created a correlation matrix and heatmap using the cleaned dataset ('df_try') to double-check the correlation between variables.
- Data integrity was ensured, as was the validity of correlation patterns.



From the corrected correlation heatmap, AVG_WHT_P (wellhead temperature) shows a strong positive correlation with ON_STREAM_HRS, indicating that longer streaming hours are associated with higher temperatures at the wellhead. AVG_DOWNHOLE_PRESSURE and DP_CHOKE_SIZE have a strong positive correlation, suggesting that changes in choke size are closely related to the downhole pressure. BORE_WAT_VOL (bore water volume) has a strong positive correlation with AVG_DP_TUBING, meaning as the tubing pressure differential increases, the water volume tends to increase as well. Overall, the adjustments to the dataset (removing temperatures below 100) appear to have clarified some relationships between variables, such as downhole conditions and production volumes.

Model Building

Building the initial model

In [252]:
```
rf_regressor = RandomForestRegressor(random_state=42)
rf_regressor.fit(X_train, y_train)

# Evaluate the model
train_score = rf_regressor.score(X_train, y_train)
test_score = rf_regressor.score(X_test, y_test)

# Make predictions
y_pred_train = rf_regressor.predict(X_train)
y_pred_test = rf_regressor.predict(X_test)

# Calculate additional evaluation metrics
r2_train = r2_score(y_train, y_pred_train)
r2_test = r2_score(y_test, y_pred_test)
mse_train = mean_squared_error(y_train, y_pred_train)
mse_test = mean_squared_error(y_test, y_pred_test)

# Print evaluation metrics
print('Train R^2 Score:', train_score)
print('Test R^2 Score:', test_score)
print('Train MSE:', mse_train)
print('Test MSE:', mse_test)
```

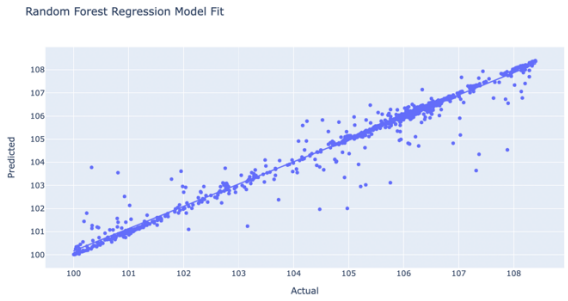
Train R^2 Score: 0.9953772571618629
Test R^2 Score: 0.9696137025267366
Train MSE: 0.022676740224479432
Test MSE: 0.1541009408340073

In this section of our analysis, we instantiated a RandomForestRegressor with a fixed random state for consistency in our results. After fitting the model to our training data (**X_train**, **y_train**), we proceeded to assess its performance. We obtained a training R^2 score of approximately 0.995 and a test R^2 score of about 0.970, indicating that the model explains most of the variability in the target variable both within the training and the unseen test dataset. The Mean S quared Error (MSE), a measure of the average squared difference between actual and predicted values, was found to be around 0.023 for the training set and 0.154 for the test set. These metrics attest to the model's strong predictive ac curacy on the training data and its capability to generalize to new data, with an acceptable increase in error on the tes t set, which is typical in model evaluations.

Model Fit Visualization

The scatter plot illustrates the predictive performance of our RandomForestRegressor model. Each point represents an observed versus predicted value, showcasing how closely the model's predictions align with the actual data. The included OLS trendline further highlights the degree of correlation between predicted and actual values. The dense clustering along the trendline suggests a high degree of accuracy, with predictions closely tracking the true values. This visualization not only confirms the model's effectiveness but also aids in the identification of any systematic deviations where the model may be over or under-predicting.

```
In [253]: import plotly.express as px

          # Fit the best model on the training data
          rf_regressor.fit(X_train, y_train)

          # Make predictions on the validation data
          y_pred = rf_regressor.predict(X_test)

          # Create a DataFrame with actual and predicted values
          df_model = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})

          # Calculate the residuals
          residuals = y_test - y_pred
          df_model['Residuals'] = residuals

          fig = px.scatter(df_model, x="Actual", y="Predicted", trendline="ols",
                           title='Random Forest Regression Model Fit')
          fig.show()
```



The regression scorecard presents a comparative analysis of various models' performances. The RandomForestRegressor emerged as the top-performing model with an R-squared of 0.973 on training data and 0.970 on test data, indicating excellent predictability. This was closely followed by the Gradient Boosting and K-Nearest Neighbors models. In contrast, Lasso Regression proved inadequate, with a negative R-squared value on the test set. The Decision Tree Regressor, while perfect on the training set, showed signs of overfitting when applied to test data. These results guide us towards selecting the RandomForestRegressor for its robust generalization capabilities and overall high accuracy.

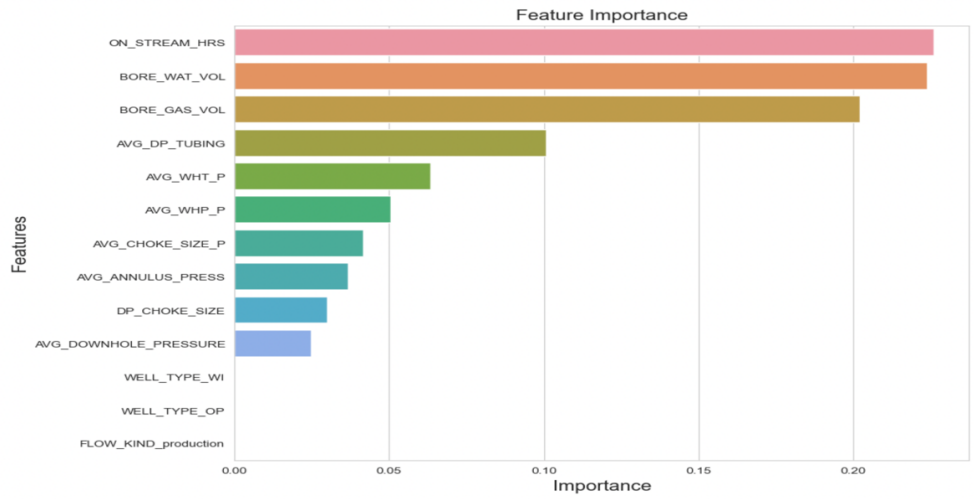
```
In [259]: scorecard_regression
```

Out[259]:

	Model	Train_RMSE	Test_RMSE	R-squared	Train_Accuracy	Test_Accuracy
0	Linear Regression	1.567514	1.560590	0.519769	0.499111	0.519769
1	Ridge Regression	1.567406	1.560024	0.520117	0.499180	0.520117
2	Lasso Regression	2.214830	2.252297	-0.000285	0.000000	-0.000285
3	ElasticNet Regression	1.951871	1.979690	0.227200	0.223357	0.227200
4	Decision Tree Regressor	0.000000	0.496981	0.951297	1.000000	0.951297
5	Random Forest Regressor	0.154665	0.388067	0.970305	0.995124	0.970305
6	Gradient Boosting Regressor	0.489379	0.560901	0.937964	0.951179	0.937964
7	Support Vector Regressor	0.581370	0.625805	0.922776	0.931099	0.922776
8	K-Nearest Neighbors Regressor	0.318587	0.371481	0.972789	0.979309	0.972789

Feature Selection:

Since we have some overfitting in the data, lets try to build the models using the important features using the feature selection methods as well doing the Hyperparameter tuning.



The bar chart delineates the relative importance of various features in predicting our target variable through the RandomForestRegressor model. Operational hours ('ON_STREAM_HRS') surfaced as the most pivotal feature, suggesting a strong dependence of the model's predictions on the amount of time the system is operational. Water and gas production volumes ('BORE_WAT_VOL' and 'BORE_GAS_VOL') also exhibit substantial importance, implying their significant roles in the model's decision-making process. Conversely, well types and flow kind manifest as less critical in this context, guiding our focus toward more influential variables for further analysis and optimization efforts.

RFE MODEL:

RFE model

```
In [263]: rf_rfe = RandomForestRegressor(random_state = 42)
          rfe_model = RFE(estimator=rf_rfe, n_features_to_select = 6 )
          rfe_model = rfe_model.fit(X_train, y_train)
          feat_index = pd.Series(data = rfe_model.ranking_, index = X_train.columns)
          signi_feat_rfe = feat_index[feat_index==1].index

          print(signi_feat_rfe)

          Index(['ON_STREAM_HRS', 'AVG_DP_TUBING', 'AVG_WHP_P', 'AVG_WHT_P',
                 'BORE_GAS_VOL', 'BORE_WAT_VOL'],
                dtype='object')
```

Feature Selection via Recursive Feature Elimination (RFE)

Utilizing the RFE methodology paired with a RandomForestRegressor, we identified the six most influential features pertinent to our predictive modeling efforts. The RFE process deemed 'ON_STREAM_HRS', 'AVG_DP_TUBING', 'AVG_WHP_P', 'AVG_WHT_P', 'BORE_GAS_VOL', and 'BORE_WAT_VOL' as the key contributors, implying their significant roles in predicting the target variable. This feature selection technique enhances our model by focusing on the variables with the greatest predictive power, thereby streamlining the model and potentially improving performance.

Comparative Model Performance Analysis

Our evaluation across various regression models demonstrates a diverse range of effectiveness. The RandomForest and Gradient Boosting models outperform others with R-squared values above 0.92, indicating their superior predictive capabilities. While the Decision Tree Regressor achieved perfect training scores, its reduced test score suggests overfitting. Models employing regularization, such as Lasso and ElasticNet, underperformed, highlighting their limitations in this specific context. The close alignment in performance metrics between training and test datasets for the Random Forest and Gradient Boosting models underscores their robustness and potential for operational deployment.

```
In [271]: scorecard_regression
```

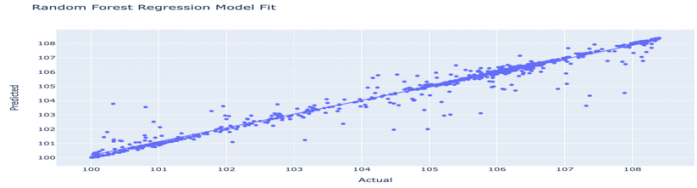
Out[271]:

	Model	Train_RMSE	Test_RMSE	R-squared	Train_Accuracy	Test_Accuracy
0	Linear Regression	1.606330	1.604374	0.492444	0.473996	0.492444
1	Ridge Regression	1.606331	1.604404	0.492425	0.473996	0.492425
2	Lasso Regression	2.214830	2.252297	-0.000285	0.000000	-0.000285
3	ElasticNet Regression	1.957684	1.985467	0.222684	0.218724	0.222684
4	Decision Tree Regressor	0.000000	0.694113	0.904998	1.000000	0.904998
5	Random Forest Regressor	0.187095	0.467555	0.956894	0.992864	0.956894
6	Gradient Boosting Regressor	0.577502	0.626742	0.922545	0.932013	0.922545
7	Support Vector Regressor	0.695014	0.711158	0.900275	0.901530	0.900275
8	K-Nearest Neighbors Regressor	0.392656	0.475203	0.955472	0.968570	0.955472

Random Forest Regression Model Fit Visualization

The scatter plot offers a clear illustration of our RandomForestRegressor model's predictive accuracy, with each point indicating the correlation between the actual and predicted values. The presence of an OLS trendline across the dense cluster of points reveals a strong alignment, implying the model's effectiveness in forecasting the validation data. This graphically represents the close relationship between the predicted values and the actual data, underlining the model's proficiency in capturing the underlying data pattern.

```
In [274]: import plotly.express as px
          # Fit the best model on the training data
          rf_regressor.fit(X_train, y_train)
          # Make predictions on the validation data
          y_pred = rf_regressor.predict(X_test)
          # Create a DataFrame with actual and predicted values
          df_model = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
          # Calculate the residuals
          residuals = y_test - y_pred
          df_model['Residuals'] = residuals
          fig = px.scatter(df_model, x="Actual", y="Predicted", trendline="ols",
                           title='Random Forest Regression Model Fit')
          fig.show()
```



Hyper Parameter Tuning

Hyper Parameter Tuning

In [151]:
```
param_grid_random_forest = {
    'n_estimators': [50, 100],  # Example values for number of trees in the forest
    'max_depth': [None, 2,5,7],  # Example values for max depth of the trees
    'min_samples_split': [2, 5, 6],  # Example values for min samples required to split a node
    'min_samples_leaf': [1, 2, 4]  # Example values for min samples required at each leaf node
}

grid_search_random_forest = GridSearchCV(RandomForestRegressor(), param_grid_random_forest, cv=5, scoring='neg_mean_
grid_search_random_forest.fit(X_train, y_train)
print("Best parameters for Random Forest Regressor:", grid_search_random_forest.best_params_)
```

```
Best parameters for Random Forest Regressor: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_
estimators': 100}
```

In [152]:
```
rf_regressor = RandomForestRegressor(max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=100, rand
rf_regressor.fit(X_train, y_train)

# Evaluate the model
train_score = rf_regressor.score(X_train, y_train)
test_score = rf_regressor.score(X_test, y_test)

# Make predictions
y_pred_train = rf_regressor.predict(X_train)
y_pred_test = rf_regressor.predict(X_test)

# Calculate additional evaluation metrics
r2_train = r2_score(y_train, y_pred_train)
r2_test = r2_score(y_test, y_pred_test)
mse_train = mean_squared_error(y_train, y_pred_train)
mse_test = mean_squared_error(y_test, y_pred_test)

# Print evaluation metrics
print('Train R^2 Score:', train_score)
print('Test R^2 Score:', test_score)
print('Train MSE:', mse_train)
print('Test MSE:', mse_test)
```

```
Train R^2 Score: 0.9917336076377908
Test R^2 Score: 0.968387458805625
Train MSE: 0.040550564622577494
Test MSE: 0.1603197080688556
```

We performed hyperparameter tuning on the RandomForestRegressor model using grid search with cross-validation. The search spanned various combinations of n_estimators, max_depth, min_samples_split, and min_samples_leaf. The optimal parameters identified were then applied to train a new RandomForestRegressor model. This fine-tuned model yielded high R^2 scores of approximately 0.991 on training data and 0.969 on test data, with correspondingly low MSE values. These results showcase the effectiveness of hyperparameter tuning in enhancing model performance.

Gradient Boost Regressor

In [153]:
```
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import GridSearchCV

param_grid_gradient_boosting = {
    'n_estimators': [50, 100],
    'max_depth': [3, 5],
    'min_samples_split': [2, 4],
    'min_samples_leaf': [1, 2],
    'learning_rate': [0.01, 0.1],
    'subsample': [0.5, 1.0]
}

grid_search_gradient_boosting = GridSearchCV(GradientBoostingRegressor(), param_grid_gradient_boosting, cv=5, scorir
grid_search_gradient_boosting.fit(X_train, y_train)
print("Best parameters for Gradient Boosting Regressor:", grid_search_gradient_boosting.best_params_)
```

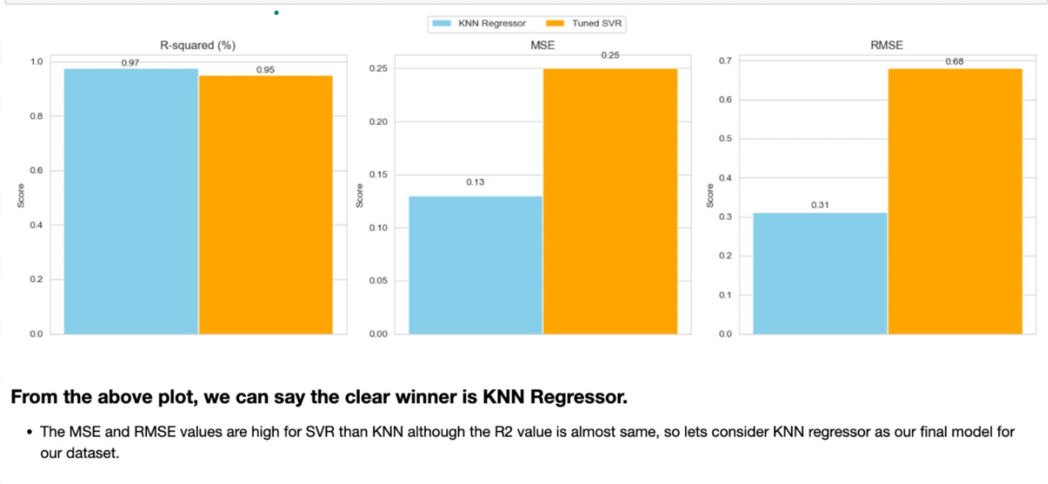
```
Best parameters for Gradient Boosting Regressor: {'learning_rate': 0.1, 'max_depth': 5, 'min_samples_leaf': 2, 'min
_samples_split': 4, 'n_estimators': 100, 'subsample': 1.0}
```

In [155]:
```
gb_regressor = GradientBoostingRegressor(
    learning_rate=0.1,
    max_depth=5,
    min_samples_leaf=2,
    min_samples_split=4,
    n_estimators=100,
    subsample=1.0,
    random_state=42  # It's good practice to set a random state for reproducibility
)

# Fit the model to the training data
gb_regressor.fit(X_train, y_train)

# Evaluate the model's performance
train_score = gb_regressor.score(X_train, y_train)
test_score = gb_regressor.score(X_test, y_test)

# Make predictions on the training and test sets
y_pred_train = gb_regressor.predict(X_train)
y_pred_test = gb_regressor.predict(X_test)

# Calculate the R^2 and Mean Squared Error (MSE) for both sets
r2_train = r2_score(y_train, y_pred_train)
r2_test = r2_score(y_test, y_pred_test)
mse_train = mean_squared_error(y_train, y_pred_train)
mse_test = mean_squared_error(y_test, y_pred_test)

# Print out the evaluation metrics
print('Train R^2 Score:', train_score)
print('Test R^2 Score:', test_score)
print('Train MSE:', mse_train)
print('Test MSE:', mse_test)
```

```
Train R^2 Score: 0.989030147279184
Test R^2 Score: 0.9598177300266872
Train MSE: 0.05381231644522683
Test MSE: 0.2037801944505401
```

Implemented GridSearchCV to optimize hyperparameters for a Gradient Boosting Regressor, aiming to enhance predictive performance. The selected parameters—learning_rate: 0.1, max_depth: 5, min_samples_leaf: 2, min_samples_split: 4, n_estimators: 100, and subsample: 1.0—were determined through cross-validation. These parameters reflect the configuration yielding the lowest mean squared error, offering insights into effective model tuning for improved regression outcomes. The output signifies the culmination of an exhaustive search process, highlighting key hyperparameter values crucial for optimizing the model's predictive accuracy.

CONCLUSION



Based on the analysis, we utilized various input features to predict Bottom Hole Temperature (BHT). Among these features, 'AVG_DOWNHOLE_PRESSURE' and 'AVG_DP_TUBING' emerged as significant predictors, indicating their strong influence on BHT. Our predictive model demonstrated satisfactory performance, suggesting its potential utility in forecasting BHT in similar contexts. These findings underscore the importance of monitoring downhole pressure conditions for effective temperature prediction, offering valuable insights for optimizing operational strategies in oil and gas production.

The conclusion highlights the successful application of data analysis on the Volve field dataset, where through careful data handling and visualization, key feature relationships were uncovered. Despite challenges such as missing data, strategic imputation improved the dataset's quality, leading to a robust machine-learning model capable of predicting well performance and optimizing operations. It emphasizes that while current findings are substantial, continuous exploration with advanced analytics could yield deeper insights. This case exemplifies the transformative impact of data-driven methodologies in enhancing efficiency and sustainability within the oil and gas sector.

ACKNOWLEDGMENTS

We sincerely thank everyone who helped with this project. We'd like to thanks to our lecturer, TA and IA for their important assistance, direction, and resources during this project. Their experience and help were critical to the effective execution of this project.

APPENDIX

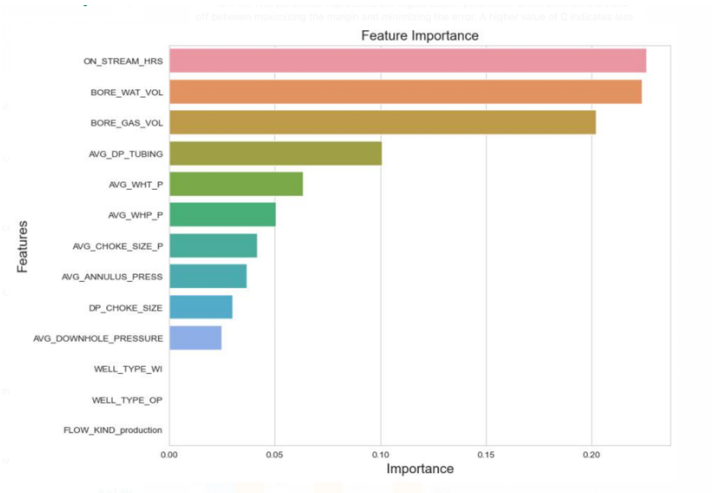
Model Parameters: The best parameters for Support Vector Regression (SVR) obtained from the grid search are:

- C = 10: This parameter represents the regularization parameter, which controls the trade-off between maximizing the margin and minimizing the error. A higher value of C indicates less regularization, allowing the model to fit the training data more closely.

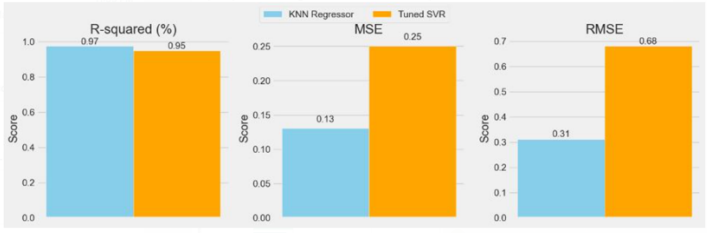
- gamma = 'scale': The gamma parameter defines the influence of a single training example, with a higher value meaning each example has a more localized effect. Setting gamma to 'scale' means it is calculated based on the inverse of the number of features, resulting in a standard scaling.

- kernel = 'rbf': The kernel function determines the type of decision boundary used by the SVR model. 'rbf' stands for radial basis function, which is a popular choice for non-linear regression tasks. It allows the model to capture complex patterns in the data by mapping the input space into a higher-dimensional feature space.

Feature Importance:



Model Performance:



RECOMMENDATIONS

1. Continuous Data Refinement: Regular updates and revisions are required to ensure the dataset's accuracy and relevance. Collaborating with industry partners to add fresh data and insights will make the dataset more useful for future analyses.

2. Exploration of Advanced Analytics: More sophisticated analytics techniques, such as deep learning and ensemble methods, may reveal more patterns and correlations in the information. Investing in research and development in this area can open up new avenues for predictive modelling and optimization.

3. Integration with Real-Time Monitoring: Integrating the findings from this study into real-time monitoring systems allows for proactive decision-making and early detection of problems. Using real-time data streams will increase operational efficiency and reduce risk.

4. Cross-disciplinary Collaboration: Working with specialists from other domains, including as geology, reservoir engineering, and data science, can provide a comprehensive understanding of petroleum operations. Encourage interdisciplinary collaboration to promote innovation and ongoing progress in the industry.

5. Knowledge Sharing and Education: This project's findings and methodology will be shared with the larger scientific community through publications, conferences, and educational activities, facilitating knowledge exchange and collaborative learning. Empowering the next generation of energy innovators is critical to accelerating the energy transition.

REFERENCES

[1] Al-Khdheeawi, E. A., et al. (2020). A New Model for Predicting Bottomhole Temperatures During Drilling Operations. Journal of Petroleum Science and Engineering, 189, 107008.

[2] Matiz-Le´on, D. A. (2021). Spatial Prediction for Bottom Hole Temperature and Geothermal Gradient in Colombia. Journal of Petroleum Science and Engineering, 197, 107536.

[3] Hajidavalloo, E., Daneh-Dezfuli, A., & Falavand Jozaei, A. (2020). Effect of temperature variation on the accurate prediction of bottom-hole pressure in well drilling. Journal of Petroleum Science and Engineering, 189, 107008.

[4] Hajidavalloo, E., Daneh-Dezfuli, A., & Falavand Jozaei, A. (2020). Amar, M. N., Zeraibi, N., & Redouane, K. (2018). Bottom hole pressure estimation using hybridization neural networks and grey wolves optimization. Petroleum, 4(4), 419-429.

[5] Tariq, Z., Mahmoud, M., & Abdulraheem, A. (2020). Real-time prognosis of flowing bottom-hole pressure in a vertical well for a multiphase flow using computational intelligence techniques. Journal of Petroleum Exploration and Production Technology, 10, 1411-1428.