ABSTRACT

Oil production forecasting is a crucial aspect of the oil and gas industry, as it enables decision-makers to plan and optimize production, investment, and exploration strategies. Traditional forecasting methods have limitations in terms of accuracy and speed, which is why machine learning has emerged as a promising solution to improve the forecasting process. In this paper, we present a comprehensive review of the state-of-the-art machine learning techniques used for oil production forecasting. We discuss the data sources and pre-processing techniques used to prepare data for machine learning models. We also review the most commonly used machine learning algorithms, including linear regression, decision trees, random forests, support vector machines, artificial neural networks, and deep learning. We present case studies and examples of oil production forecasting using machine learning, including predicting production from existing wells, forecasting production from new wells, and predicting oil reserves. We also discuss the challenges and limitations of machine learning for oil production forecasting, such as data quality, model complexity, and interpretability. Our review shows that machine learning is a powerful tool for oil production forecasting, with the potential to significantly improve the accuracy and efficiency of the process. However, it is important to carefully select appropriate data sources, pre-processing techniques, and machine learning algorithms to ensure the best results. Further research is needed to address the challenges and limitations of machine learning for oil production forecasting and to develop more advanced and accurate models.

TABLE OF CONTENTS:

2. Introduction:

Machine learning is a subfield of artificial intelligence (AI) that involves developing algorithms and statistical models that enable computers to learn from and make predictions or decisions based on data, without being explicitly programmed to do so. In other words, it is a method of teaching machines to learn patterns and relationships in data, and use this knowledge to make predictions or decisions about new data.

There are three main types of machine learning algorithms: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the algorithm is trained on a labeled dataset, where the correct output is known for each input. The algorithm learns to map inputs to outputs, and can then make predictions on new, unseen data. In unsupervised learning, the algorithm is trained on an unlabeled dataset, and must find patterns or relationships in the data on its own. Reinforcement learning is a type of learning in which an algorithm learns to make decisions based on feedback it receives from its environment.

Machine learning is used in a wide range of applications, including image recognition, speech recognition, natural language processing, recommendation systems, and predictive analytics. It has become an increasingly important tool in many industries, including healthcare, finance, and transportation, among others.

Oil production forecasting using machine learning is important for several reasons:

2.1 Improved decision making:

Accurate forecasting of oil production can help companies and policymakers make better decisions about production levels, pricing, and investment in new technologies. Machine learning algorithms can help to identify patterns in historical production data, which can be used to make more accurate predictions about future production levels.

2.2 Cost savings:

Accurate forecasting can help companies to optimize their production processes, reduce waste, and avoid costly shutdowns or overproduction. By using machine learning to analyze production data, companies can identify inefficiencies and areas for improvement, which can result in significant cost savings.

2.3 Increased efficiency:

Machine learning algorithms can help companies to predict equipment failures, optimize maintenance schedules, and improve overall efficiency. By identifying patterns in sensor data, for example, algorithms can predict when equipment is likely to fail, allowing maintenance to be scheduled in advance, which can reduce downtime and increase productivity.

2.4 Environmental impact:

Accurate forecasting of oil production can help companies to manage their environmental impact by reducing the risk of spills or leaks. By predicting production levels, companies can ensure that they have the necessary resources and infrastructure in place to handle the volume of oil being produced.

Overall, machine learning can help companies to optimize their production processes, reduce costs, and improve their environmental impact, making it a valuable tool for the oil and gas industry.

3. Data Description:

The Volve oil field dataset is a comprehensive dataset of production and reservoir data from the Volve oil field, located in the Norwegian North Sea. The dataset was released by Equinor (formerly Statoil), a Norwegian energy company, and includes data from the production period of the Volve oil field from 2008 to 2016. The features in the dataset are

- Onstreaminject hrs
- BORE WI VOL
- ON STREAM HRS
- AVG DP TUBING
- AVG ANNULUS PRESS
- AVG CHOKE SIZE P
- DP CHOKE SIZE
- WHT P and WHP P
- BORE OIL GAS WATER VOLUME

3.1 Onstreaminject hrs:

"Onstreaminject_hrs" refers to the number of hours that injection wells are in operation or online in an oil field.

Injection wells are used in oil and gas production to inject fluids such as water, gas, or chemicals into the reservoir to maintain pressure and improve recovery rates. The term "onstream" generally refers to the time that a system or equipment is in operation or online, while "inject" obviously refers to the process of injecting fluids into the reservoir. "Hrs" likely stands for "hours", indicating that the data refers to the number of hours that injection wells are operational or online.

Therefore, "Onstreaminject_hrs" is likely a data point related to the operational performance of injection wells in an oil field, specifically the number of hours that they are in operation.

3.2 BORE WI VOL:

The "BORE WI VOL" most likely refers to "Borehole Water Injection Volume". This term is used to describe the volume of water that is injected into a borehole or well during water injection operations.

Water injection is a process used in the oil and gas industry to increase the pressure in a reservoir and push hydrocarbons towards production wells. This process involves pumping water into the reservoir through injection wells, which are separate from production wells. The volume of water injected can be an important performance metric for tracking the efficiency and effectiveness of water injection operations.

Therefore, "BORE WI VOL" likely represents the amount of water that is injected into a borehole or well during water injection operations, and is a data point that is relevant for oilfield operations and management.

3.3 ON STREAM HRS:

In the oil and gas industry, "ON STREAM HRS" most likely refers to "On-stream Hours". This term is used to describe the number of hours that a particular piece of equipment or system is in operation or online, such as a production well, pipeline, or facility.

The "on-stream" generally refers to the time during which a particular asset or system is actively contributing to production or processing operations. Therefore, "ON STREAM HRS" would represent the total number of hours that a particular asset or system is actively contributing to oil and gas production or processing.

Tracking on-stream hours is an important performance metric in oilfield operations and management, as it allows operators to monitor the uptime and efficiency of production assets and identify opportunities for optimization and improvement. By monitoring on-stream hours, operators can identify equipment or systems that may require maintenance or repairs, as well as areas where production can be increased or optimized.

In the context of the oil and gas industry, "Average downhole pressure and temperature" are two important measurements used to monitor and optimize well performance.

The downhole refers to the portion of the wellbore that is below the surface and where the production zone or reservoir is located. The average downhole pressure is the average pressure in the reservoir or production zone, and is typically measured in pounds per square inch (psi) or bar. This measurement is important because it indicates the pressure that the reservoir is exerting on the production well, and therefore the potential flow rate of hydrocarbons from the reservoir into the wellbore. By monitoring the average downhole pressure, operators can optimize production by adjusting the flow rate and other parameters to maintain a consistent pressure and flow rate.

The average downhole temperature, on the other hand, is the average temperature in the reservoir or production zone. This measurement is important because it affects the viscosity of the hydrocarbons in the reservoir, which in turn affects the flow rate and production rate of the well. As the temperature increases, the viscosity of the hydrocarbons decreases, which can improve flow and production rates. However, excessive temperatures can also cause issues such as corrosion and scaling, so it's important to monitor and control the downhole temperature within a certain range to optimize production while minimizing operational issues.

By measuring and monitoring the average downhole pressure and temperature, operators can gain valuable insights into the performance of their wells and optimize production to maximize recovery rates and profitability.

3.4 AVG DP TUBING:

The"AVG DP TUBING" most likely refers to "Average Differential Pressure in the Tubing".

Tubing is the pipe that runs from the wellhead to the bottom of the wellbore, and is used to transport hydrocarbons and other fluids to the surface. Differential pressure refers to the difference in pressure between two points, and in this case, it likely refers to the difference in pressure between the bottom of the tubing and the surface of the wellhead.

The "AVG DP TUBING" data point would represent the average differential pressure in the tubing over a certain period of time, and is typically measured in pounds per square inch (psi) or bar. This measurement is important because it can indicate the flow rate of hydrocarbons from the reservoir to the surface, as well as the presence of any blockages or restrictions in the tubing. A higher differential pressure could indicate higher flow rates or the presence of restrictions, while a lower differential pressure could indicate lower flow rates or the absence of restrictions.

By monitoring the average differential pressure in the tubing, operators can optimize production by adjusting the flow rate, identifying and addressing blockages or restrictions, and ensuring that the well is operating efficiently and effectively.

3.5 AVG ANNULUS PRESS:

"AVG ANNULUS PRESS" most likely refers to "Average Annulus Pressure".

The annulus refers to the space between the tubing and the casing in a wellbore. The casing is the larger diameter pipe that lines the wellbore and provides structural support, while the tubing is the smaller diameter pipe that runs inside the casing and carries hydrocarbons to the surface. The annulus is the space between these two pipes and can be used for a variety of purposes, such as for injecting fluids into the wellbore or monitoring pressure.

The "AVG ANNULUS PRESS" data point would represent the average pressure in the annulus over a certain period of time, and is typically measured in pounds per square inch (psi) or bar. This measurement is important because it can indicate the pressure in the wellbore and provide insights into the behavior of the reservoir and the efficiency of the well.

3.6 AVG CHOKE SIZE P:

The "AVG CHOKE SIZE P" most likely refers to "Average Choke Size Percentage".

The choke valve is a key component in controlling the flow rate of hydrocarbons from the wellbore to the surface. It is typically located near the wellhead and can be adjusted to regulate the flow of fluids by restricting or allowing flow through the valve. Choke size refers to the opening size of the valve and is usually measured in fractions of an inch or millimeters.

The "AVG CHOKE SIZE P" data point represents the average choke size percentage over a certain period of time. This measurement is typically expressed as a percentage of the total available choke opening size. For example, if the total choke opening size is 2 inches and the average choke size percentage is 50%, then the choke valve was open to an average size of 1 inch during the monitoring period.

The choke size percentage is an important measurement because it directly affects the flow rate of hydrocarbons from the wellbore to the surface. By adjusting the choke size, operators can regulate the flow rate and optimize production while maintaining safe and efficient operations.

By monitoring the average choke size percentage, operators can identify potential issues such as equipment malfunction, wellbore blockages, or changes in reservoir behavior. Additionally, changes in choke size percentage can be used to optimize production and recovery rates. Therefore, "AVG CHOKE SIZE P" is an important data point in oilfield operations and management.

In the oil and gas industry, "DP CHOKE SIZE" most likely refers to "Differential Pressure across the Choke Valve".

The choke valve is a critical component in controlling the flow rate of hydrocarbons from the wellbore to the surface. It is located near the wellhead and can be adjusted to regulate the flow of fluids by restricting or allowing flow through the valve. Differential pressure refers to the difference in pressure between two points, in this case, the pressure upstream and downstream of the choke valve.

3.7 DP CHOKE SIZE:

The "DP CHOKE SIZE" data point represents the differential pressure across the choke valve over a certain period of time and is typically measured in pounds per square inch (psi) or bar. This measurement is important because it indicates the pressure drop across the valve and directly affects the flow rate of hydrocarbons from the wellbore to the surface.

By monitoring the differential pressure across the choke valve, operators can adjust the valve opening to maintain the desired flow rate while ensuring safe and efficient operations. Changes in differential pressure can also be used to identify potential issues such as equipment malfunction or blockages in the wellbore.

Therefore, "DP CHOKE SIZE" is an important data point in oilfield operations and management, as it provides key information about the flow rate and efficiency of hydrocarbon production from the wellbore.

3.8 WHT P and WHP P:

The wellhead is the surface equipment that is installed at the top of a wellbore to control the flow of fluids from the well. Wellhead temperature and pressure are important parameters in oil and gas production because they can affect the behavior of the fluids in the reservoir and the production rate of the well.

Wellhead pressure refers to the pressure of the fluids at the wellhead, which is measured in pounds per square inch (psi) or bar. Wellhead temperature refers to the temperature of the fluids at the wellhead, which is measured in degrees Celsius (°C) or degrees Fahrenheit (°F).

WHT P is often measured and recorded by sensors and instruments installed at the wellhead, and this data is used by engineers and production personnel to monitor and optimize the production of the well. The average WHT P value is the average of the wellhead temperature and pressure over a specific time period and can be used to gain insights into the behavior of the reservoir and the performance of the well.

As mentioned earlier, the wellhead is the surface equipment that is installed at the top of a wellbore to control the flow of fluids from the well. Wellhead pressure refers to the pressure of the fluids at the wellhead, which is measured in pounds per square inch (psi) or bar.

3.9 BORE OIL GAS WATER VOLUME:

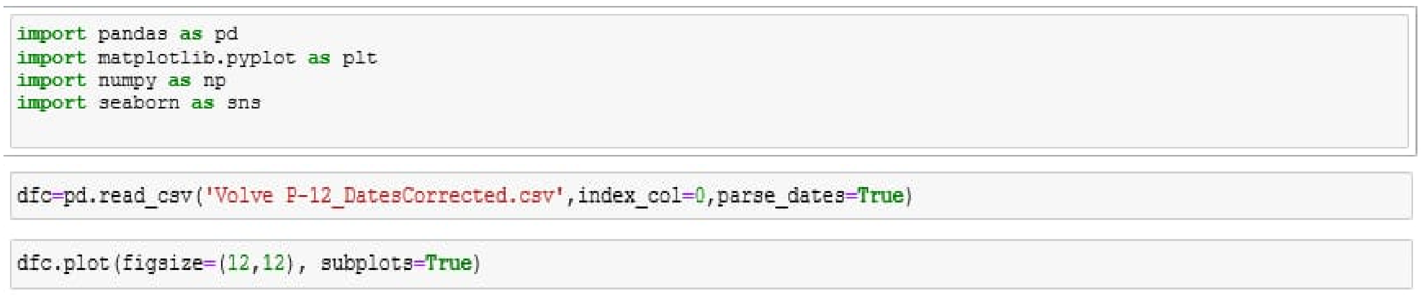
"BORE OIL GAS WATER VOLUME" is a data point that provides information about the volume of oil, gas, and water produced from a wellbore over a certain period of time.

Oil, gas, and water are the three main types of hydrocarbons that can be extracted from the subsurface. The amount of each fluid produced can vary depending on the characteristics of the reservoir, the well completion design, and other factors.

The "BORE OIL GAS WATER VOL wellbore during the monitoring period. This measurement is typically expressed in barrels or cubic meters and can be used to calculate production rates and evaluate the efficiency of the well.

By monitoring the volume of oil, gas, and water produced from the wellbore, operators can identify potential issues such as fluid migration, wellbore damage, or changes in reservoir behavior. Additionally, changes in the relative amounts of each fluid produced can be used to optimize production and recovery rates.

Therefore, "BORE OIL GAS WATER VOLUME" is an important data point in oilfield operations and management, as it provides key information about the productivity and efficiency of hydrocarbon production from the wellbore.



```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

```
dfc=pd.read_csv('Volve P-12_DatesCorrected.csv',index_col=0,parse_dates=True)
```

```
dfc.plot(figsize=(12,12), subplots=True)
```

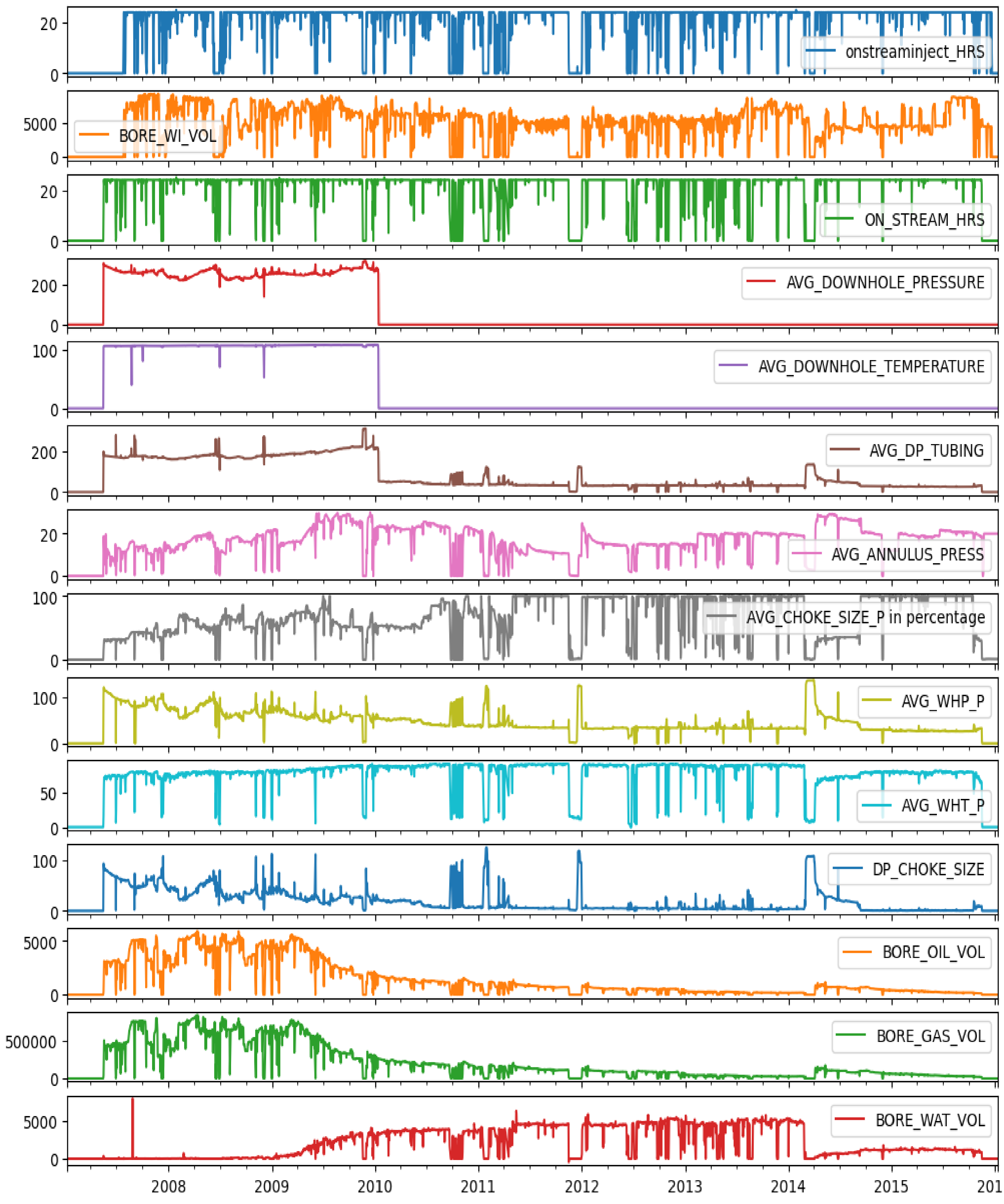


Figure 1. Data Visualization of Volve oil field dataset

4. Methodology:

4.1 Missing Values or Null values in Dataset:

4.2 Uses of finding null in data in ML:

Finding null or missing values in data is an essential task in Machine Learning (ML) because null values can impact the performance of the ML models. Here are some of the uses of finding null values in data in ML. They are

- Data Cleaning
- Feature Selection
- Better Data Analysis
- Improve Model Performance
- Prepare Data for ML Algorithms

4.2.1 Data Cleaning:

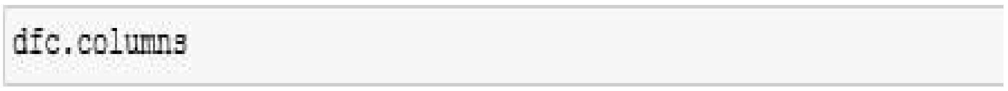
Null values in data can cause errors in machine learning models, and thus it is essential to find and eliminate them. By identifying the missing values in the data, we can perform data cleaning and impute the missing values using various methods such as mean, median, mode, or regression.

4.2.2 Feature Selection:

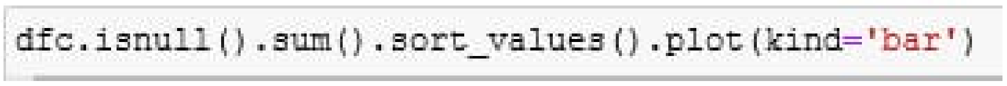
Identifying the features that contain a large number of missing values can help us decide whether to keep or drop those features. We can also consider the impact of the missing values on the overall accuracy of the model and decide accordingly.

4.2.3 Better Data Analysis:

By identifying the null values in the data, we can gain insights into the quality and completeness of the data. We can determine whether the missing values are random or not, and we can use this information to design better ML models.



```
dfc.columns
```



```
dfc.isnull().sum().sort_values().plot(kind='bar')
```

4.2.4 Improve Model Performance:

The presence of null values can lead to biased predictions and affect the performance of ML models. By identifying and imputing the missing values, we can improve the performance of our models and get more accurate predictions.

4.2.5 Preparing Data for ML Algorithms:

Some ML algorithms cannot handle missing values, and thus we need to impute them before feeding them into the algorithms. By identifying the missing values, we can prepare the data for the ML algorithms by imputing them appropriately.

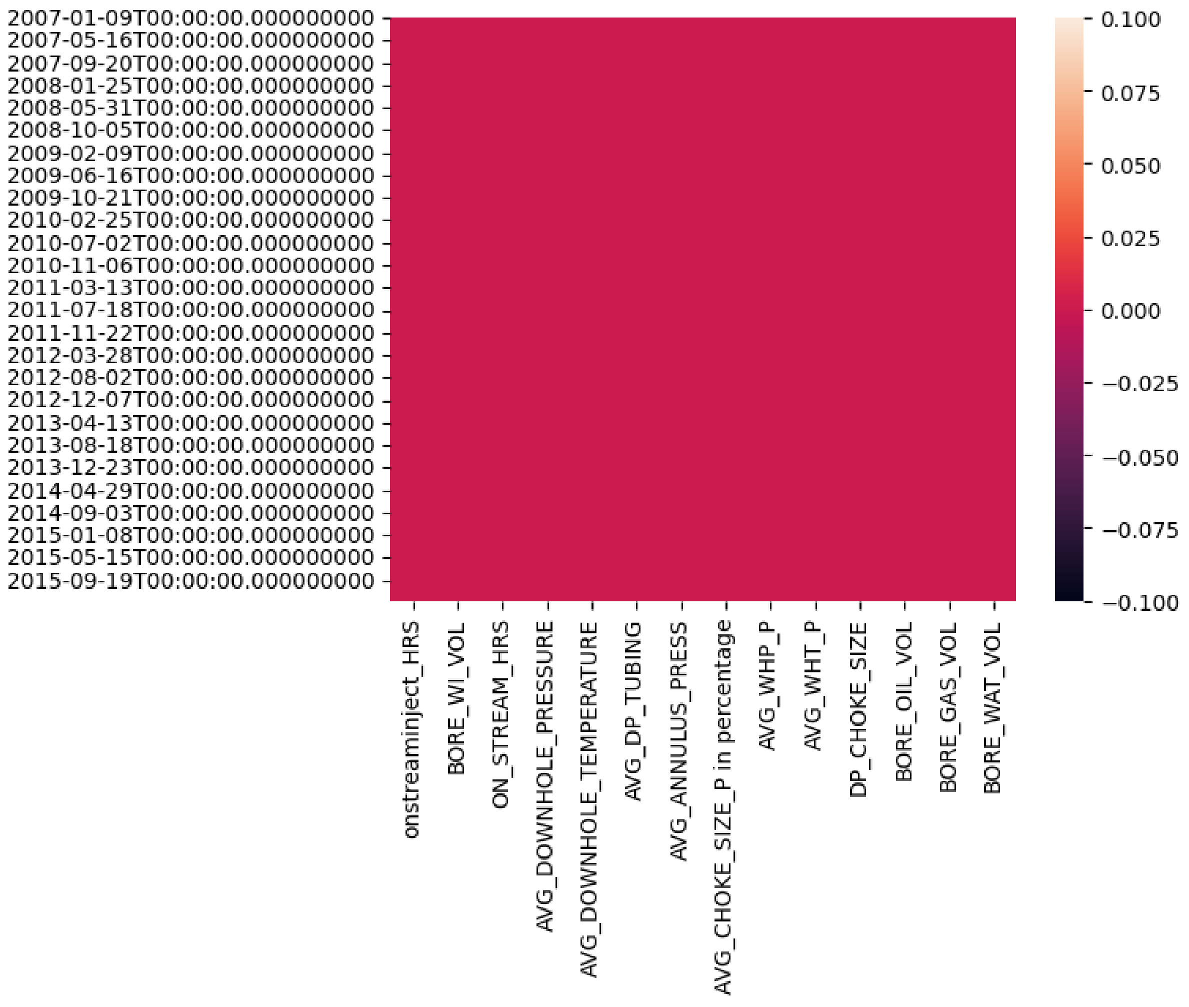


Figure 2. Data Visualization of Null Values

4.2.6 Correlations of data:

Finding correlation among data is a crucial task in Machine Learning (ML) as it helps us understand the relationship between different features in the data. Here are some of the uses of finding correlation among data in ML:

4.2.7 Feature Selection:

Correlation can help us identify the most significant features in the data. Highly correlated features can be redundant, and removing them can improve the performance of the ML model. Correlation analysis can help us select the most informative and non-redundant features for the ML model.

4.2.8 Data Preprocessing:

Correlation analysis can help us detect and handle missing values in the data. If two features are highly correlated, we can use one of the features to impute the missing values in the other feature. Correlation analysis can also help us identify outliers and anomalies in the data.

4.2.10 Model Selection:

Correlation analysis can help us select the appropriate ML model for the data. If the features are highly correlated, linear models such as linear regression or logistic regression may be more appropriate. If the features are not correlated or have a nonlinear relationship, other models such as decision trees, support vector machines, or neural networks may be more appropriate.

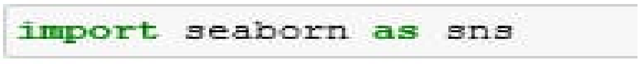
4.2.11 Model Evaluation:

Correlation analysis can help us evaluate the performance of the ML model. We can compare the correlation between the predicted values and the actual values to determine the accuracy and effectiveness of the model.

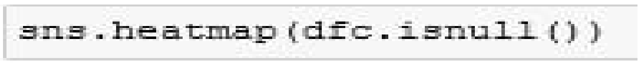
4.2.12 Data Visualization:

Correlation analysis can help us visualize the data and discover patterns and relationships between different features. We can use scatter plots or heat maps to visualize the correlation matrix and identify the most correlated features.

Overall, finding correlation among data is a critical step in the ML pipeline as it helps us understand the relationship between different features in the data, select features and models, preprocess the data, evaluate the performance of the model, and visualize the data.



```
import seaborn as sns
```



```
sns.heatmap(dfc.isnull())
```

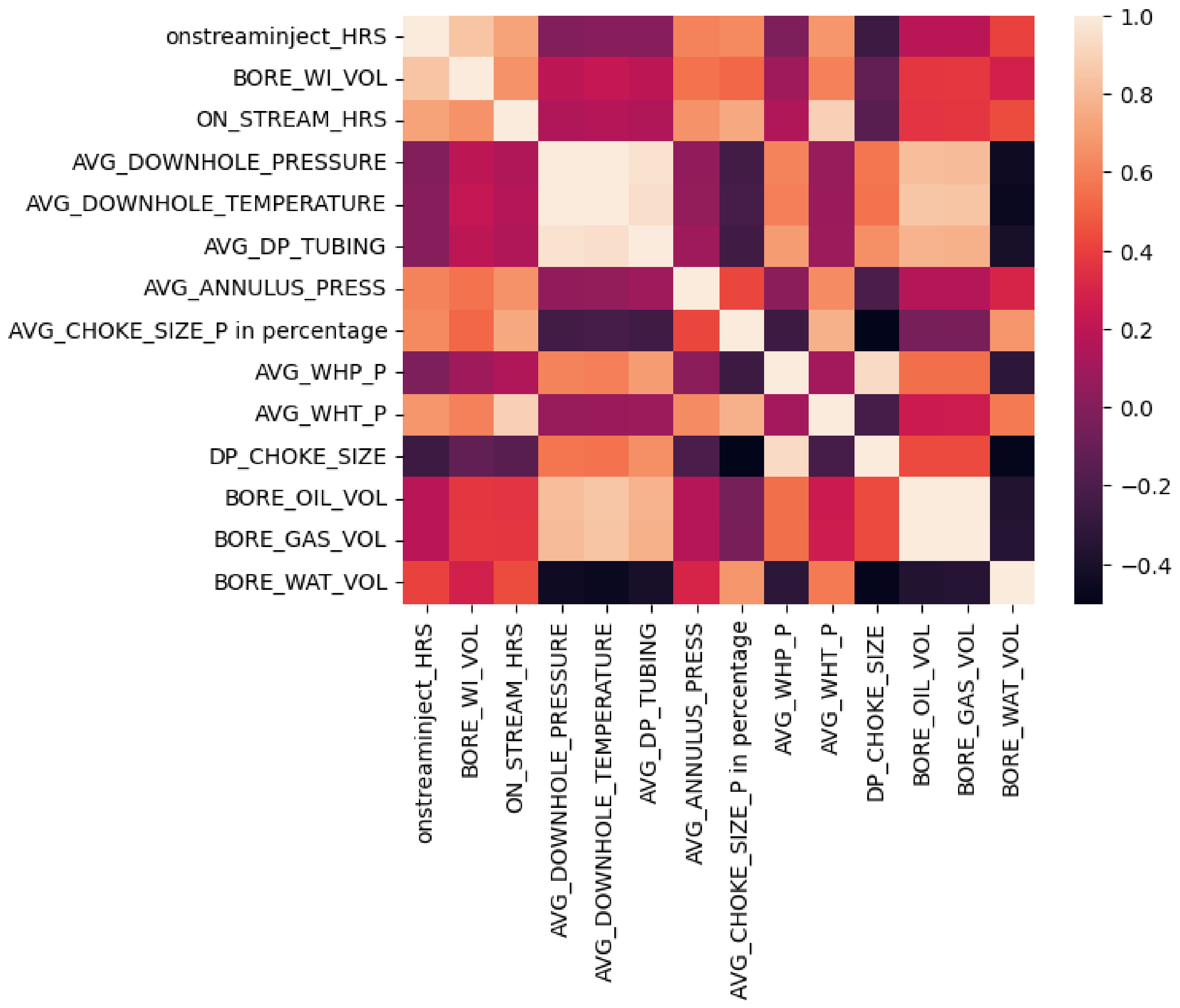


Figure 3. Data Visualization on Correlations among data

```
dfc.describe()
```

```
sns.heatmap(dfc.corr())
```

```
dfc.corr()[op_col].sort_values(ascending=False)[1:].plot(kind='bar', figsize=(12,4),grid=True)
```

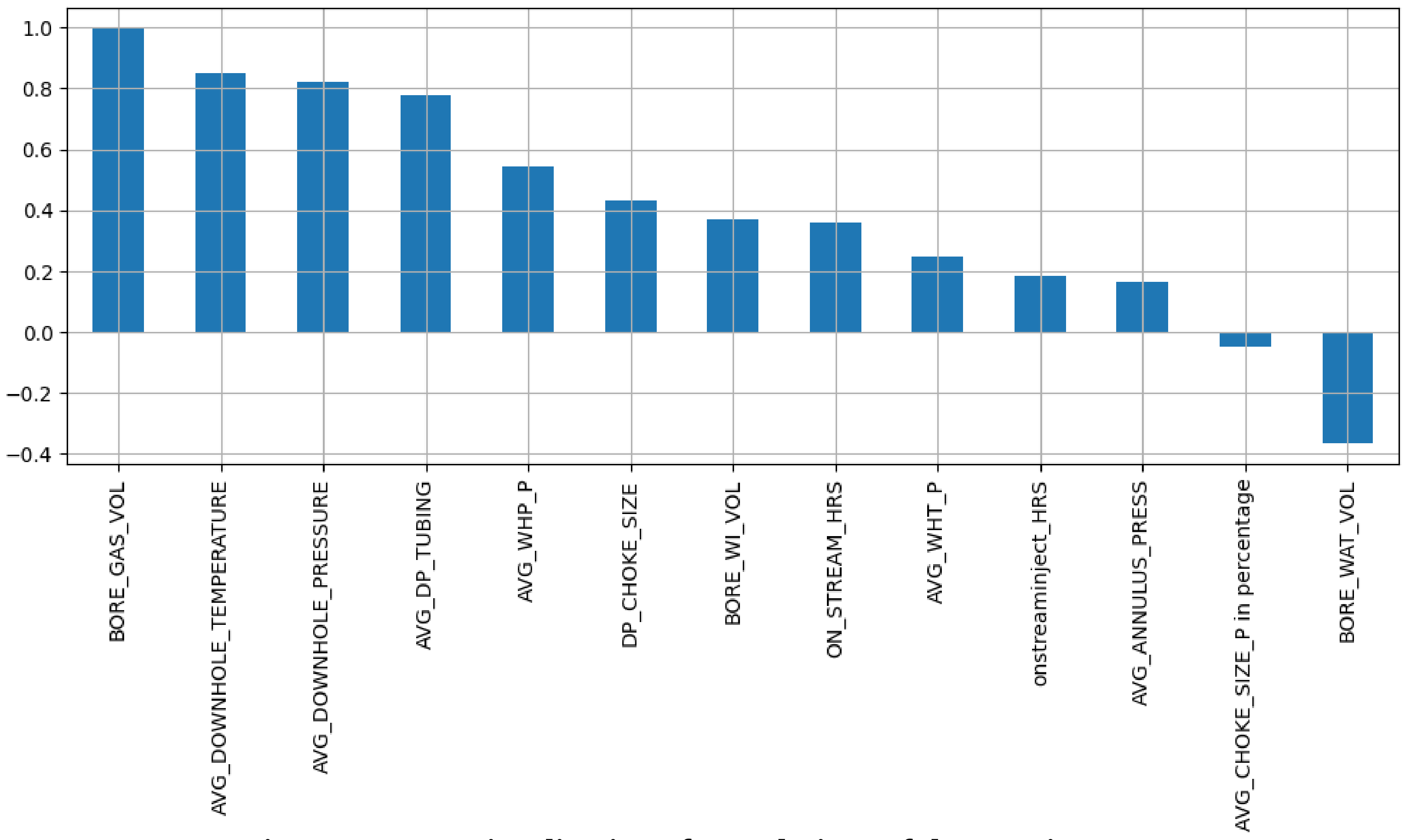


Figure 4. Data Visualization of correlations of data against Bore Oil Production Volume

4.2.13 Feature Selection:

Feature selection is the process of selecting the most informative and relevant features from a dataset that are useful for building a machine learning model. It involves identifying and removing the features that are irrelevant, redundant, or noisy, and keeping only the features that have a significant impact on the model's performance.

Feature selection can help to improve the efficiency and accuracy of a machine learning model by reducing overfitting and simplifying the model. It can also help to reduce the dimensionality of the data, making it easier to work with and visualize.

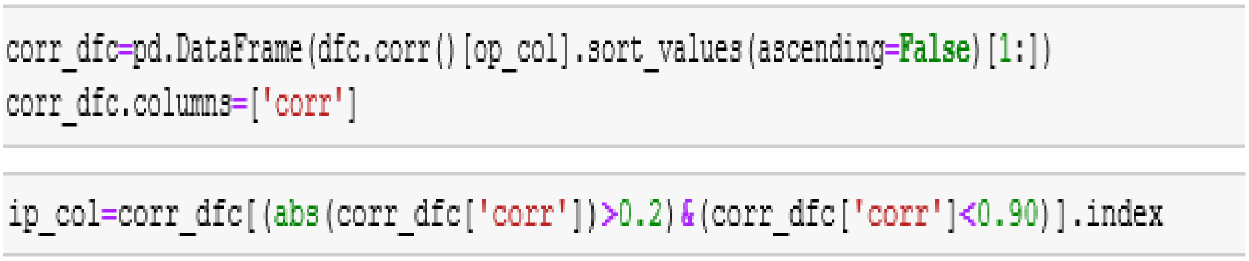
There are various methods for feature selection, including filter methods, wrapper methods, and embedded methods. Filter methods use statistical tests or other criteria to evaluate the relevance of each feature and select the top-ranked features. Wrapper methods use a machine learning algorithm to evaluate the performance of different feature subsets and select the one that performs the best. Embedded methods incorporate feature selection as part of the model training process, such as Lasso regression.

Overall, feature selection is an essential step in the machine learning pipeline, as it helps to improve the performance and efficiency of the model, and simplify the data to make it easier to work with.

Table 1:

	CORRELATION
BORE_GAS_VOL	0.998661
AVG_DOWNHOLE_TEMPERATURE	0.850241
AVG_DOWNHOLE_PRESSURE	0.81992
AVG_DP_TUBING	0.778941
AVG_WHP_P	0.544204
DP_CHOKE_SIZE	0.434375
BORE_WI_VOL	0.368739
ON_STREAM_HRS	0.361124
AVG_WHT_P	0.24575
onstreaminject_HRS	0.182405
AVG_ANNULUS_PRESS	0.163663
AVG_CHOKE_SIZE_P in percentage	-0.049757
BORE_WAT_VOL	-0.364295

Correlations of data against Bore Oil Production Volume



```
corr_dfc=pd.DataFrame(dfc.corr()[op_col].sort_values(ascending=False)[1:])
corr_dfc.columns=['corr']
```

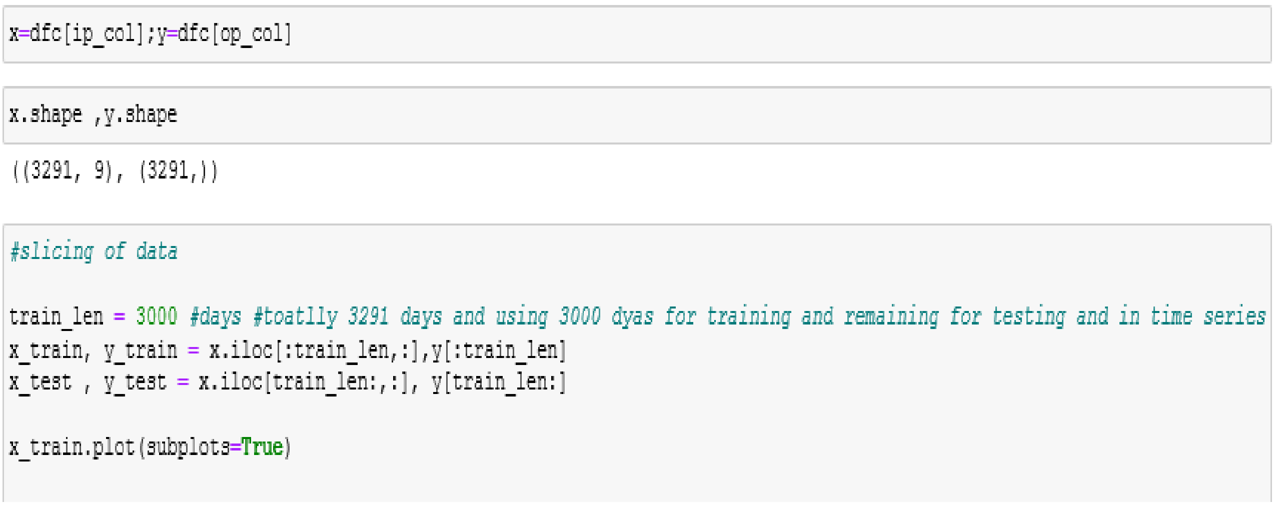
```
ip_col=corr_dfc[(abs(corr_dfc['corr'])>0.2)&(corr_dfc['corr']<0.90)].index
```

4.2.14 Data Preprocessing Techniques:

Standard Scaler is a data preprocessing technique in Machine Learning (ML) that involves scaling the features to have zero mean and unit variance. This technique is commonly used in ML to normalize the data and improve the performance of the model. Here are some of the uses of the Standard Scaler in ML:

Normalization:

Standard Scaler can help to normalize the data, making it easier to work with and visualize. By scaling the features to have zero mean and unit variance, the data is centered around zero and has a consistent scale, which can help to improve the performance of certain ML algorithms, such as linear regression or logistic regression.



```
x=dfc[ip_col];y=dfc[op_col]
```

```
x.shape ,y.shape
```

```
((3291, 9), (3291,))
```

```
#slicing of data

train_len = 3000 #days #toatlly 3291 days and using 3000 dyas for training and remaining for testing and in time series
x_train, y_train = x.iloc[:train_len,:],y[:train_len]
x_test , y_test = x.iloc[train_len:,:], y[train_len:]

x_train.plot(subplots=True)
```

4.2.15 Improving Model Performance:

Standard Scaler can help to improve the performance of the ML model by reducing the impact of the scale of the features on the model's predictions. By scaling the features, we can ensure that each feature has an equal contribution to the model, regardless of its scale. This can help to improve the accuracy and consistency of the model's predictions.

4.2.16 Reducing Overfitting:

Standard Scaler can help to reduce overfitting in the ML model by reducing the variance of the features. By scaling the features, we can ensure that the features with higher variance do not dominate the model and cause overfitting.

```
from sklearn.ensemble import RandomForestRegressor
```

```
rf=RandomForestRegressor()
rf.fit(x_train, y_train)
yp_train=rf.predict(x_train)
yp_test=rf.predict(x_test)
```

```
plt.figure(figsize=(12,4))
plt.scatter(dfc.index[:train_len],y_train, color='orange')
plt.plot(dfc.index[:train_len], yp_train)

plt.scatter(dfc.index[train_len:],y_test, color='green')
plt.plot(dfc.index[train_len:], yp_test, color='black')

plt.axvline(dfc.index[train_len], color='red')
plt.grid()
```

4.2.17 Robustness to Outliers:

Standard Scaler is robust to outliers in the data, as it is based on the distribution of the features rather than the absolute values. This can help to improve the robustness and stability of the ML model, particularly in the presence of noisy or anomalous data.

4.2.18 Process of data splitting:

Splitting the data is a critical step in the machine learning (ML) pipeline that involves dividing the dataset into two or more subsets. The main purpose of splitting the data is to train the ML model on one subset (training set) and evaluate its performance on another subset (testing set).

The most common way to split the data is to randomly divide the dataset into two subsets: the training set and the testing set. Typically, the training set contains 70-80% of the data, while the testing set contains the remaining 20-30% of the data.

During the training phase, the ML model learns the underlying patterns and relationships in the training set. Once the model is trained, it is evaluated on the testing set to measure its performance. The testing set provides an unbiased estimate of the performance of the model on new, unseen data.

It's important to note that the data splitting process must be done carefully to avoid introducing bias into the model. For example, if the data is not randomly split, the model may learn to perform well on a particular subset of the data, but fail to generalize to new data. To prevent this, we must ensure that the data is randomly split into training and testing sets, and that both sets are representative of the entire dataset.

There are 3291 observations in each features and we have splitted 3000 observations for training and 291 observations for testing our machine learning model. After splitting the training and testing data, we should standardize the splitted data for better performance of the model.

4.2.20 Model Selection:

Model selection is the process of selecting the best machine learning model among several models for a given problem. This involves defining the problem, choosing a set of candidate models, splitting the data into training and testing sets, training the models, comparing their performance, fine-tuning the selected model, and evaluating the final model. The goal is to select the model that performs the best on the testing set, while ensuring that it can generalize to new, unseen data.

The model selected for forecasting the oil production using machine learning is Random Forest Regression.

Random forest regression is a popular machine learning algorithm used for regression tasks. It is an extension of the decision tree algorithm that builds a forest of decision trees and aggregates their predictions to obtain a more accurate and robust model.

The main idea behind random forest regression is to create a large number of decision trees, each trained on a different subset of the data and using a random subset of features. During prediction, each decision tree predicts the target value, and the final prediction is the average of all the predictions from the individual trees.

Random forest regression has several advantages, including:

- It can handle both continuous and categorical variables.
- It is less prone to overfitting than a single decision tree.
- It can handle missing values and outliers in the data.
- It is relatively fast and scalable.

Random forest regression is widely used in various applications, including finance, healthcare, and marketing. It is particularly useful when the relationship between the input features and the target variable is complex and nonlinear, and when the data contains noise and outliers.

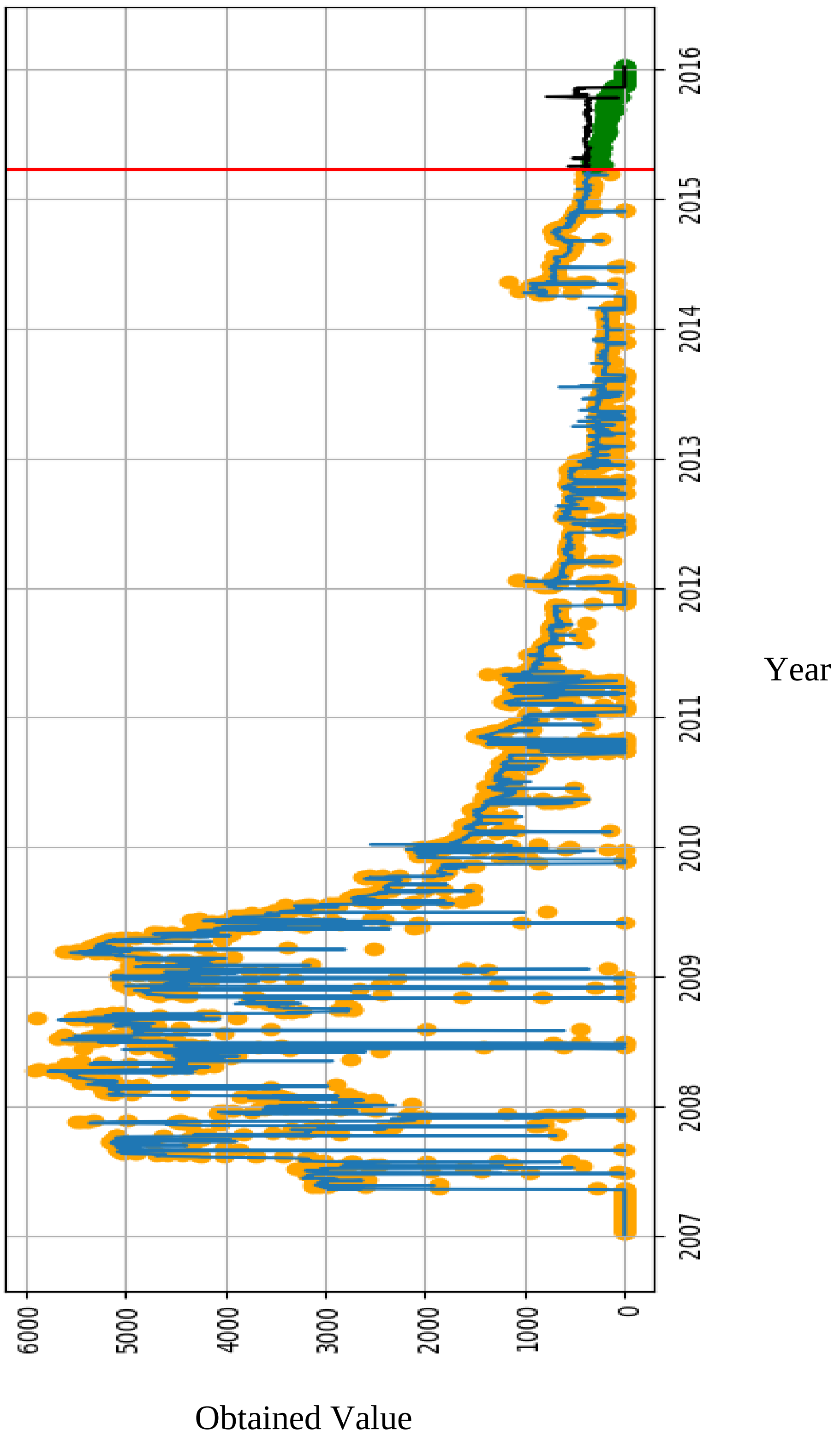


Figure 5. Data Visualization of predictions by Random Forest Regressor

5. Steps involved in Oil Forecasting:

5.1 Data Splitting:

Time series data is a type of data where each observation is recorded at a specific point in time. Machine learning models trained on time series data must take into account the temporal order of the observations. Therefore, special care must be taken when splitting time series data for training and testing.

Here are some common methods for splitting time series data for machine learning:

5.2 Training-validation-test split:

This method involves splitting the data into three parts: a training set, a validation set, and a test set. The training set is used to train the model, the validation set is used to tune hyperparameters and select the best model, and the test set is used to evaluate the performance of the final model.

5.3 Fixed window split:

This method involves dividing the time series data into fixed-length windows or blocks. The data can be split into non-overlapping windows or overlapping windows. The first part of the data is used for training, the second part is used for validation, and the remaining part is used for testing.

5.4 Sliding window split:

This method is similar to the fixed window split, but the windows slide over the time series with a certain step size. This allows for more data to be used for training and validation, but it can also increase the risk of overfitting.

5.5 Time-based split:

This method involves splitting the data based on a specific point in time. For example, you could split the data into training data from the first year, validation data from the second year, and test data from the third year.

5.6 Forecasting horizon split:

This method involves splitting the data into training data and test data based on a specific forecasting horizon. The training data is used to forecast a certain period in the future, and the test data is used to evaluate the accuracy of the forecasts.

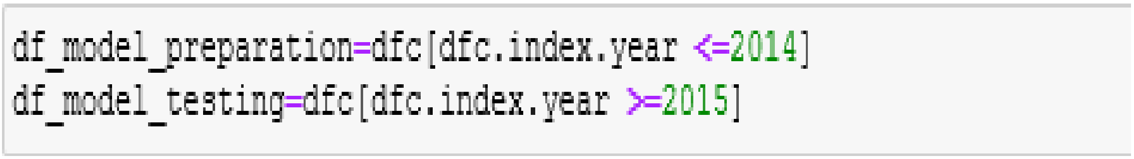
When splitting time series data for machine learning, it is important to consider the nature of the data and the problem you are trying to solve. Each method has its own advantages and disadvantages, so it is important to select the method that best suits your needs.

Here we have splitted the dataset into two, one for model training and other for model testing. Model training data contains data from 2007 to 2014 and Model testing data contains 2015 to 2016.

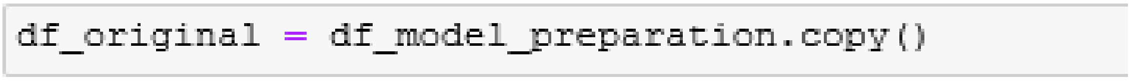
5.7 Transformation of Model Training Data:

After splitting the data, we are going to slice the bore oil volume feature alone from the model training data.

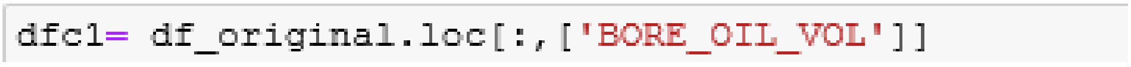
Using rolling mean function, we will transform the noisy data into cleaned data which enhances the performance of the model and helps to understand the model's behaviour easily.



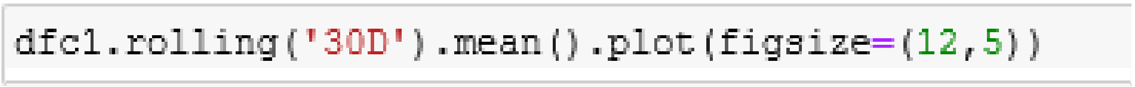
```
df_model_preparation=dfc[dfc.index.year <=2014]
df_model_testing=dfc[dfc.index.year >=2015]
```



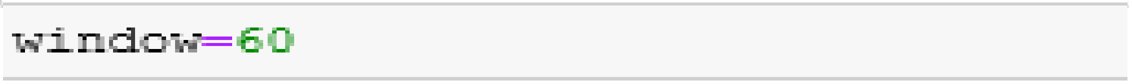
```
df_original = df_model_preparation.copy()
```



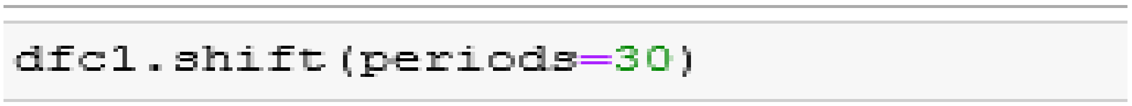
```
dfc1= df_original.loc[:,['BORE_OIL_VOL']]
```



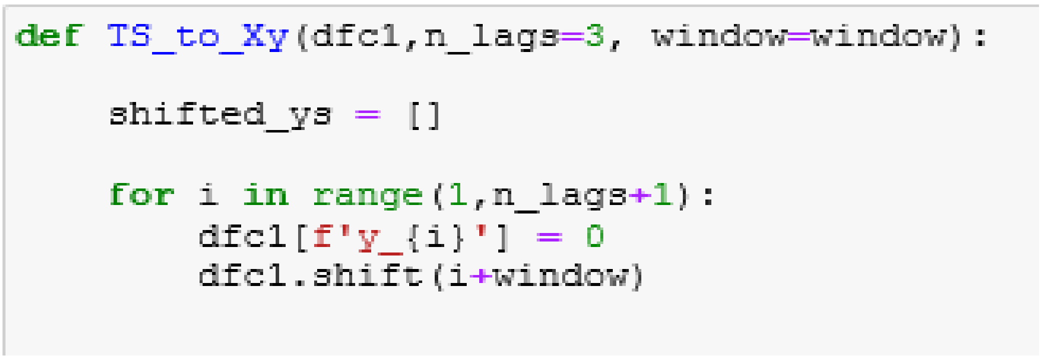
```
dfc1.rolling('30D').mean().plot(figsize=(12,5))
```



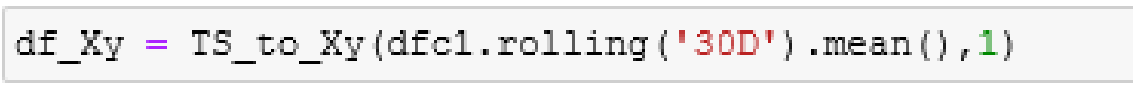
```
window=60
```



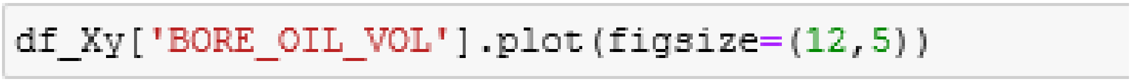
```
dfc1.shift(periods=30)
```



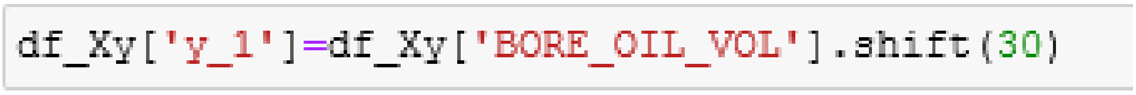
```
def TS_to_Xy(dfc1,n_lags=3, window=window):

    shifted_ys = []

    for i in range(1,n_lags+1):
        dfc1[f'y_{i}'] = 0
        dfc1.shift(i+window)
```



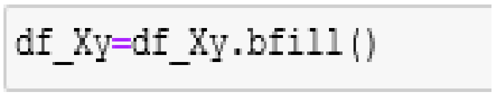
```
df_Xy = TS_to_Xy(dfc1.rolling('30D').mean(),1)
```



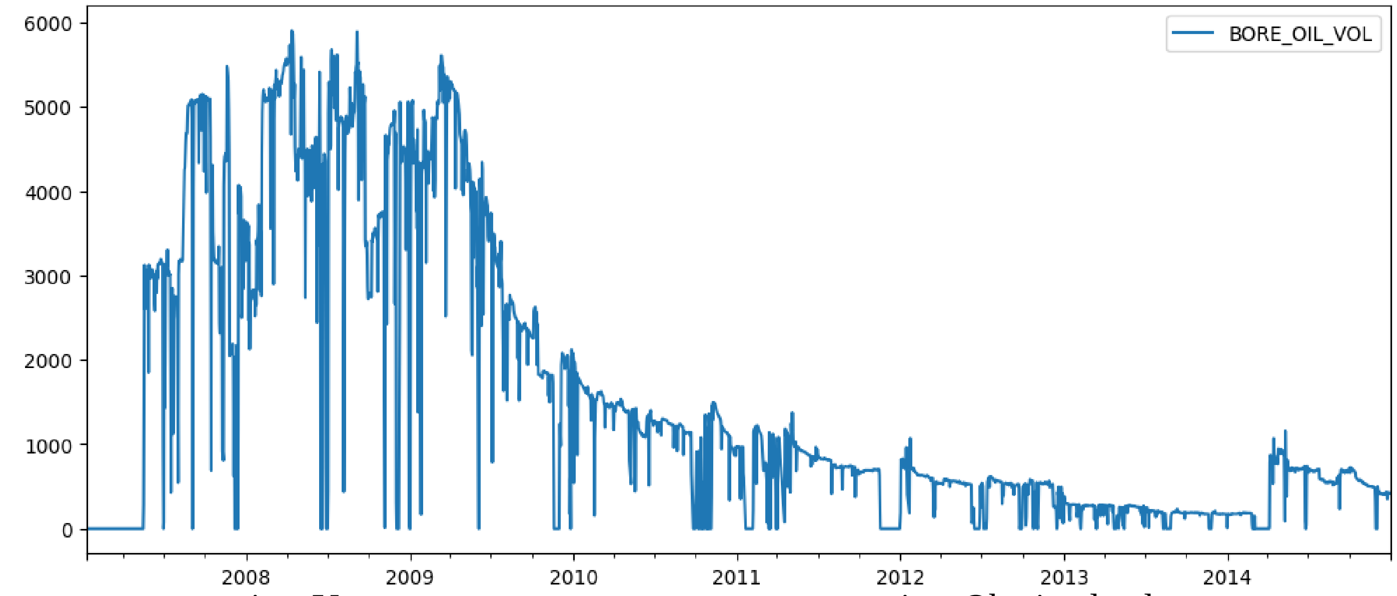
```
df_Xy['BORE_OIL_VOL'].plot(figsize=(12,5))
```



```
df_Xy['y_1']=df_Xy['BORE_OIL_VOL'].shift(30)
```

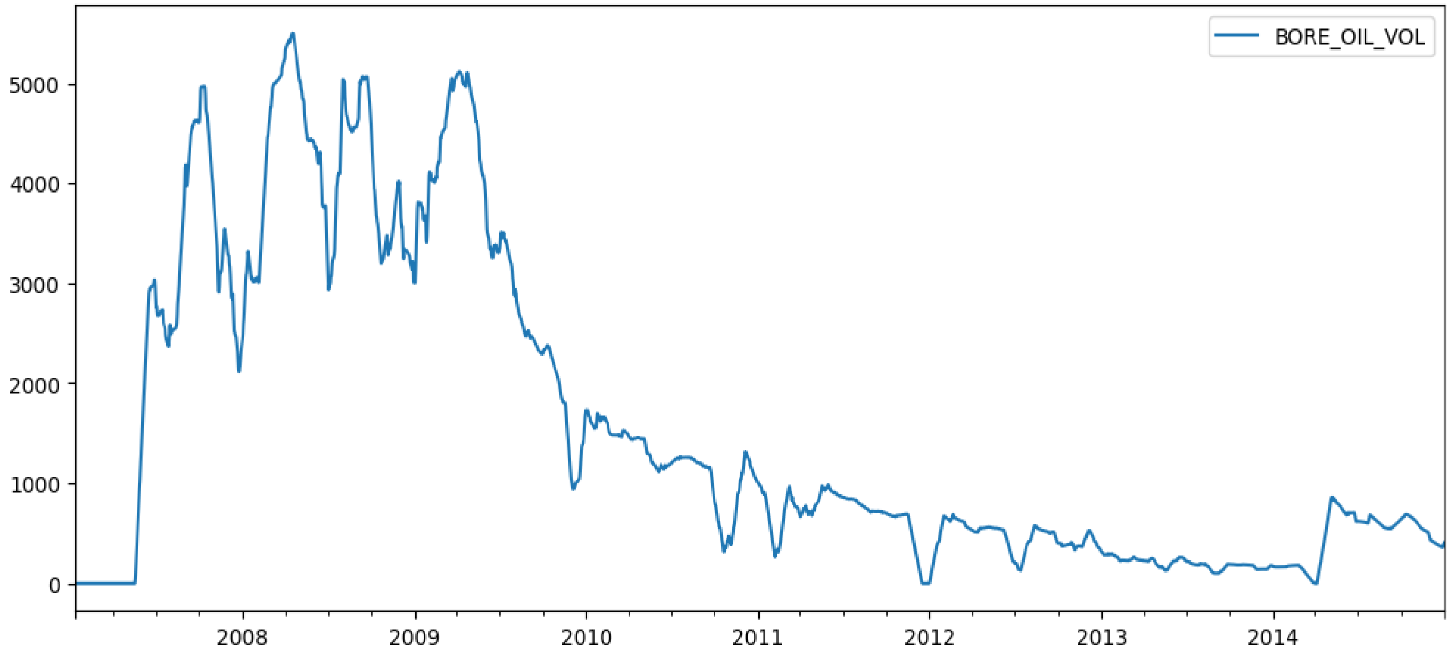


```
df_Xy=df_Xy.bfill()
```



x-axis – Year y-axis – Obtained value

Figure 6. Data Visualization of raw data before cleaning



x-axis – Year y-axis – Obtained Value

Figure 7. Data Visualization of cleaned data using Rolling Mean Function

Then we are adding the new feature ('y_1') which contains data of Bore oil volume but the data have been shifted 30 days advance.

The shifted 30 days advance because of the function of previous oil production rates will let us know the current oil production rates.

In machine learning, shifting of data and windows refers to the process of moving a sliding window over a sequence of data and analyzing the data within each window. This technique is commonly used for time-series data or sequence data.

The sliding window is a fixed-size subset of the entire data sequence, which moves along the sequence one step at a time. At each step, the data within the window is used as input to the machine learning model for analysis. This process is repeated for each window in the sequence, allowing the model to learn patterns and relationships in the data over time.

The size of the window and the step size of the sliding window are both important parameters that can affect the performance of the model. A larger window size can capture more information about the underlying patterns in the data, but may also result in increased computational complexity. A smaller window size, on the other hand, can lead to faster training times but may miss some of the finer details in the data.

The choice of window size and step size will depend on the specific task and dataset being analyzed. In some cases, it may be beneficial to use overlapping windows to capture even more information about the data. Overall, the shifting of data and windows technique can be a powerful tool for analyzing time-series and sequence data in machine learning.

Now we are going to take this two features "BORE OIL VOLUME" and "y_1" for evaluating our model. From this two features, we have taken 70 percentage of data for training and 30 percentage of data for testing.

5.8 Evaluation Metrics:

Evaluation metrics in machine learning (ML) are used to measure the performance of a model on a given dataset. The choice of evaluation metrics depends on the problem being solved and the nature of the data. Some common evaluation metrics used in ML include:

5.8.1 Accuracy: The percentage of correct predictions made by the model.

5.8.2 Precision: The proportion of true positives to the total number of positive predictions made by the model.

5.8.3 Recall: The proportion of true positives to the total number of actual positive samples in the dataset.

5.8.4 F1 score: A measure that combines both precision and recall, calculated as the harmonic mean of the two.

5.8.5 AUC-ROC: Area under the receiver operating characteristic curve, a measure of the trade-off between true positive rate and false positive rate.

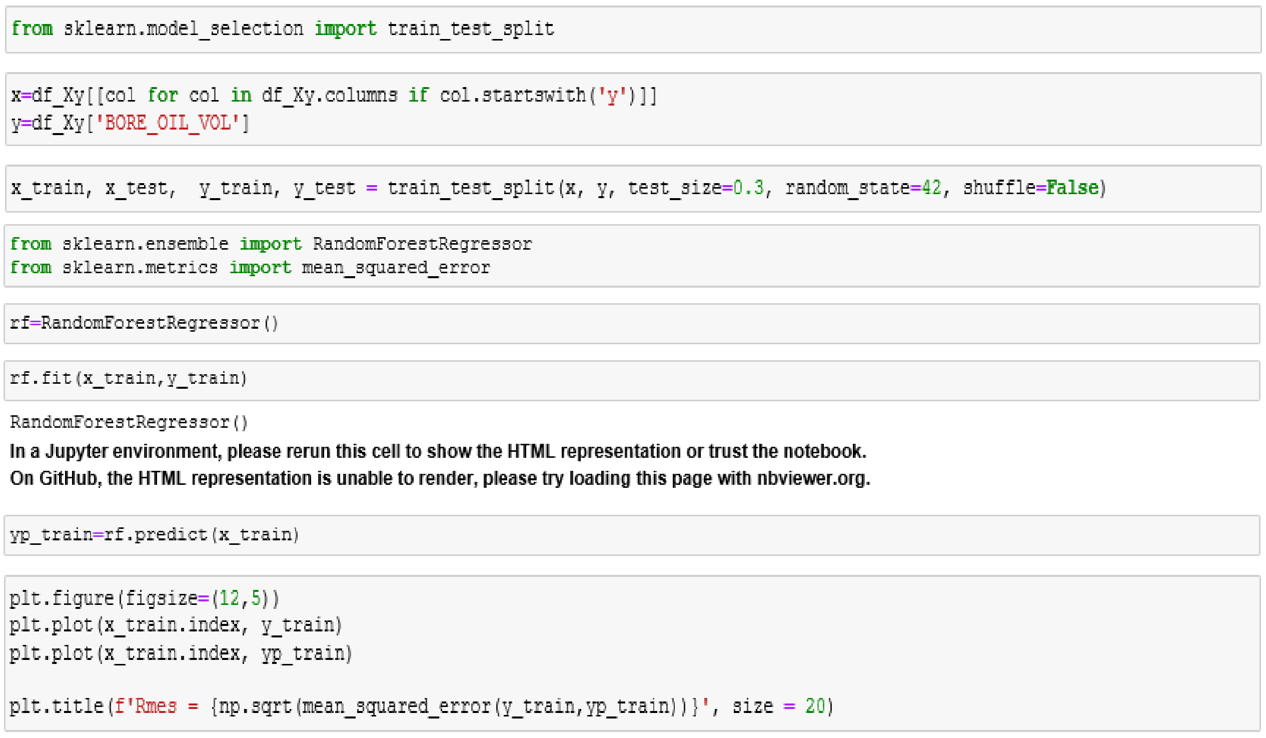
5.8.6 Mean squared error (MSE): A metric used for regression problems, which measures the average squared difference between the predicted and actual values.

5.8.7 Root mean squared error (RMSE): The square root of the MSE, which gives a measure of the average error in the predicted values.

5.8.8 Mean absolute error (MAE): A metric used for regression problems, which measures the average absolute difference between the predicted and actual values.

5.8.9 R-squared: A measure of how well the model fits the data, calculated as the proportion of variance in the dependent variable explained by the independent variables.

5.8.10 Log loss: A metric used for classification problems, which measures the performance of the model based on the predicted probability distributions.



```
from sklearn.model_selection import train_test_split
```

```
x=df_Xy[[col for col in df_Xy.columns if col.startswith('y')]]
y=df_Xy['BORE_OIL_VOL']
```

```
x_train, x_test,  y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42, shuffle=False)
```

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
```

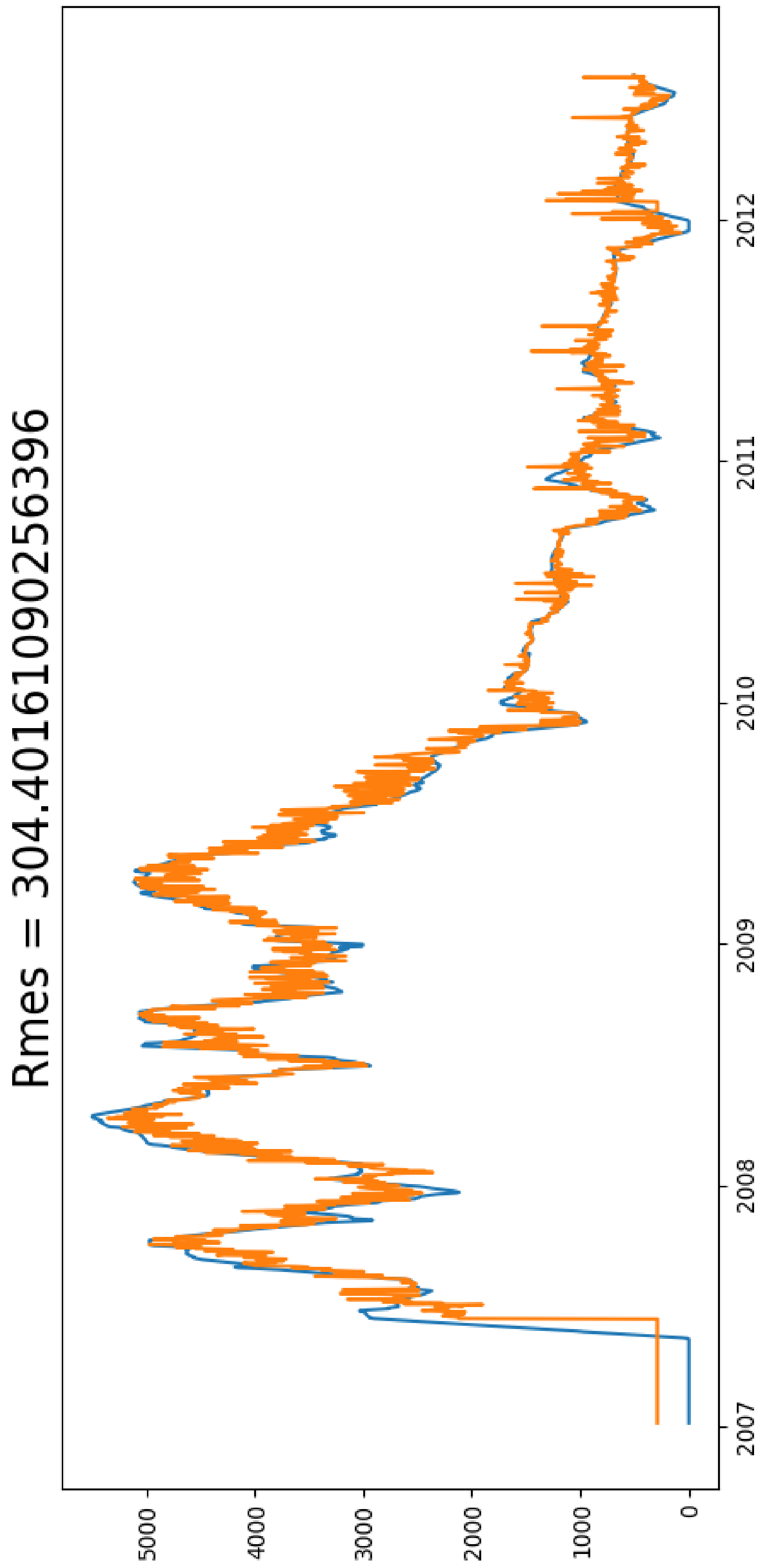
```
rf=RandomForestRegressor()
```

```
rf.fit(x_train,y_train)
```

```
RandomForestRegressor()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

```
yp_train=rf.predict(x_train)
```

```
plt.figure(figsize=(12,5))
plt.plot(x_train.index, y_train)
plt.plot(x_train.index, yp_train)

plt.title(f'Rmes = {np.sqrt(mean_squared_error(y_train,yp_train))}', size = 20)
```



x-axis – Year y-axis – Obtained Values

Figure 8. Data Visualization of Training Performance of the Machine Learning Model (Random Forest Regressor)

5.9 Decline Curve Analysis:

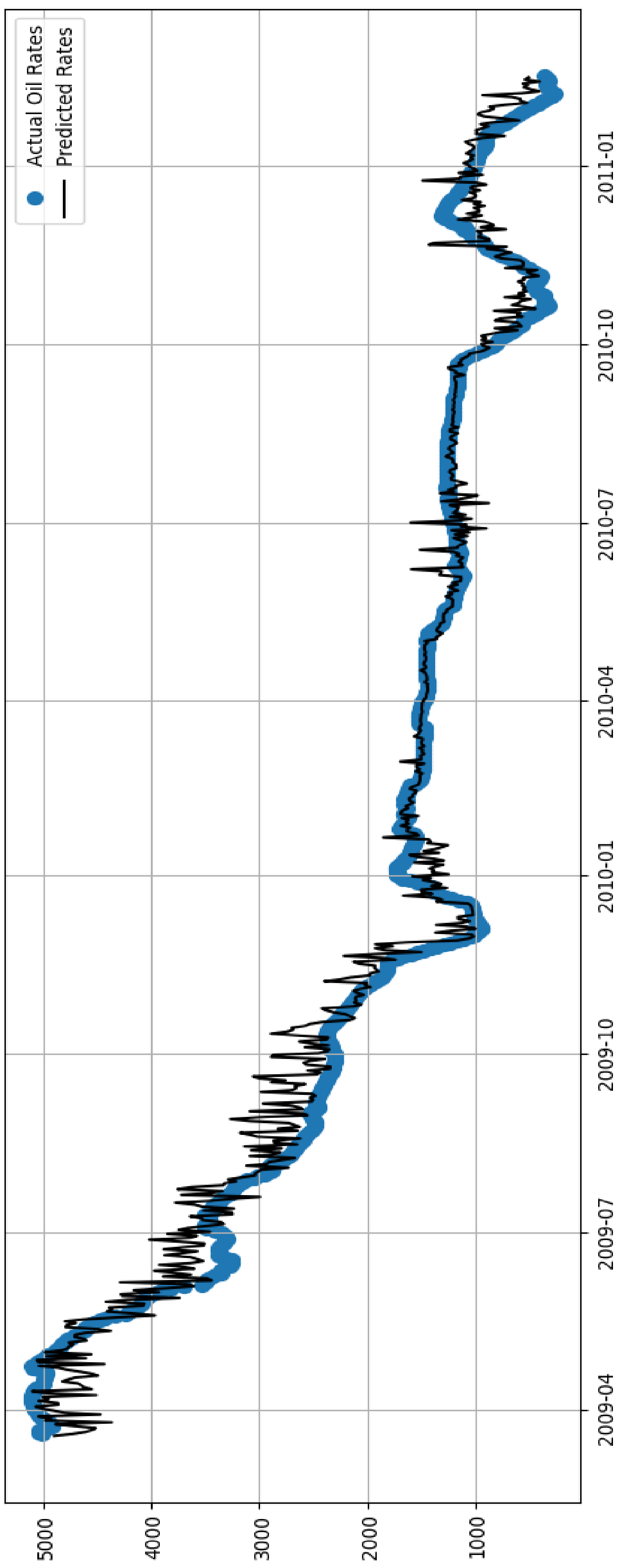
- A decline curve is a graphical representation of the decline in the production rate of an oil or gas well over time.
- The decline curve is typically plotted as a function of time on the x-axis and the production rate on the y-axis.
- The decline curve analysis is an important tool used in the oil and gas industry to estimate the future production of a well or a field, and to optimize production strategies.
- The shape of the decline curve can provide valuable information about the reservoir, such as the rate of decline and the ultimate recovery of hydrocarbons.
- There are several types of decline curves, including exponential decline, hyperbolic decline, and harmonic decline.
- Each type of decline curve has its own unique characteristics and is applicable in different reservoir scenarios.
- Overall, the decline curve is a useful tool for predicting future production and making decisions about the management of oil and gas fields.

```
ti = 800;  tf = 1500

plt.figure(figsize=(15,5))

plt.scatter(df_Xy.index[ti:tf],df_Xy['BORE_OIL_VOL'][ti:tf],marker='.',label='Actual Oil Rates',s=200)
plt.plot(df_Xy.index[ti:tf],rf.predict(df_Xy[['y_1']][ti:tf]),color='k',label='Predicted Rates')


# plt.plot(df_Xy.index[ti:tf] , df_Xy['BORE_OIL_VOL'][ti:tf], marker='x', label='Actual Oil Rates')

plt.grid()
plt.legend()
```

```
<matplotlib.legend.Legend at 0x2b32bf41490>
```

```
yp_test = rf.predict(x_test)
```

```
plt.figure(figsize=(15,5))
plt.scatter(x_test.index, y_test)
plt.plot(x_test.index, yp_test, color='orange')
```



x-axis – Year y-axis – Values Obtained

Figure 9. Data Visualization of Decline Curve in Oil Production rates

6. Model Validation:

Model validation is an essential part of machine learning (ML) that involves assessing the performance and accuracy of a trained model. Validation helps to ensure that the model is robust and reliable enough to make accurate predictions on new, unseen data.

There are several techniques used for model validation in ML, including:

6.1 Train-test split:

In this method, the available data is split into two sets - the training set and the testing set. The model is trained on the training set and then evaluated on the testing set. This approach can be useful when there is a limited amount of data available.

6.2 K-fold cross-validation:

This technique involves dividing the data into K folds or partitions. The model is trained on K-1 folds and tested on the remaining fold. This process is repeated K times, with each fold being used as the testing set once. The results are then averaged to give an overall estimate of the model's performance.

6.3 Leave-one-out cross-validation:

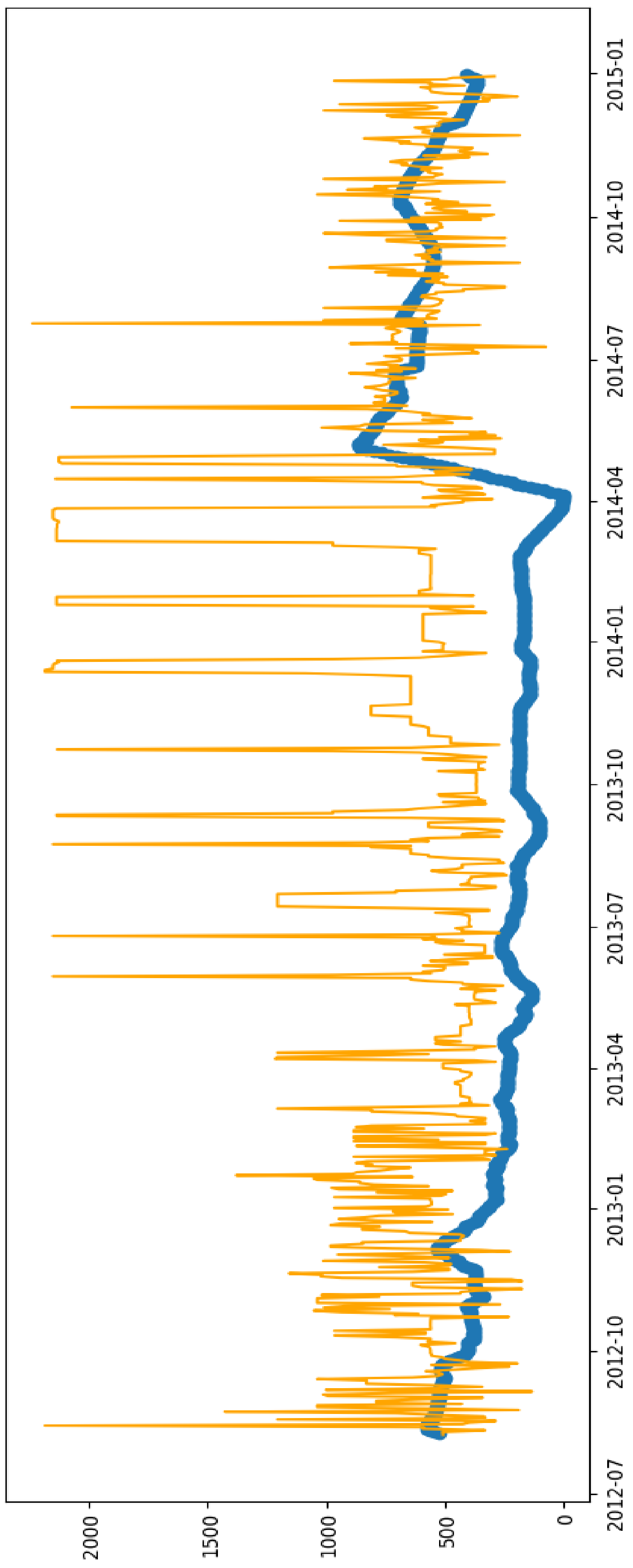
This is a type of K-fold cross-validation where K is set to the number of instances in the dataset. In this approach, the model is trained on all instances except one and then tested on the left-out instance. This process is repeated for all instances in the dataset.

6.4 Stratified sampling:

This technique is used when the data is imbalanced, meaning there are significantly more instances of one class than the other. Stratified sampling ensures that the data is split into training and testing sets in a way that preserves the relative proportions of each class.

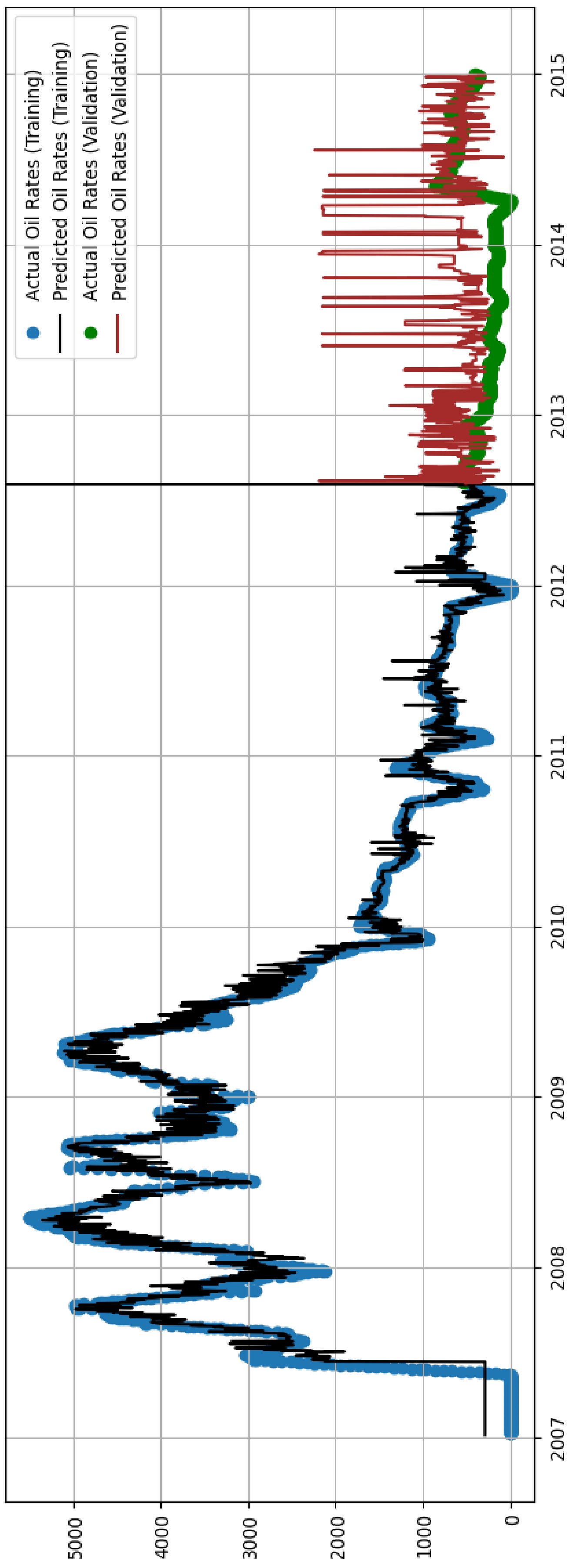
Overall, model validation is crucial for ensuring that the trained model is accurate and reliable for making predictions on new, unseen data. Different techniques can be used depending on the size and nature of the dataset and the specific requirements of the problem at hand.

```
plt.figure(figsize=(15,5))




plt.scatter(x_train.index, y_train, label='Actual Oil Rates (Training)')
plt.plot(x_train.index, yp_train, color='k', label='Predicted Oil Rates (Training)')


plt.axvline(x_test.index[0], color='k')

plt.scatter(x_test.index, y_test, color='green', label='Actual Oil Rates (Validation)')
plt.plot(x_test.index, yp_test, color='brown', label='Predicted Oil Rates (Validation)')



plt.grid()
plt.legend()
```



x-axis – Year y-axis – Values Obtained

Figure 10. Data Visualization of testing data performance by our selected model



x-axis – Year y-axis – Values Obtained

Figure 11. Data Visualization of overall performance behaviour of our model

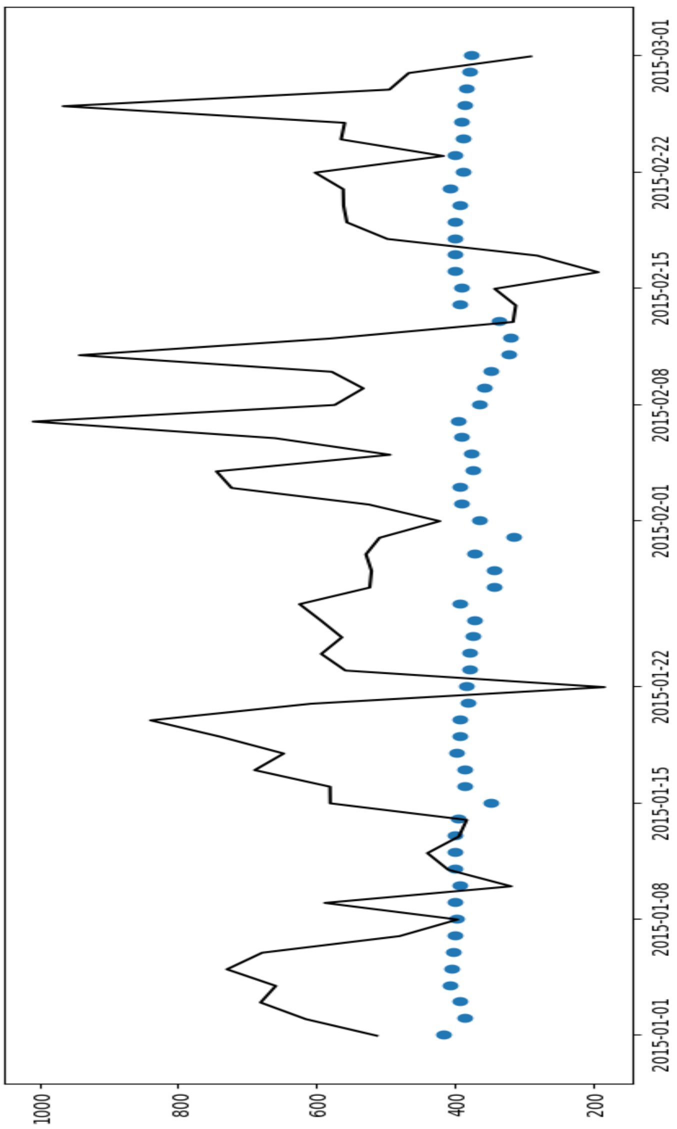
Testing of model by forecasting oil production:

```
last_input_rates = x_test.iloc[-window:,0:]

next_forecasts = np.array(rf.predict(last_input_rates))
```

```
plt.figure(figsize=(15,5))
plt.scatter(df_model_testing.index[:window], df_model_testing['BORE_OIL_VOL'][:window], label='Actual Future Rates')
plt.plot(df_model_testing.index[:window], next_forecasts.reshape(window,1), label='Model Forecasted Rates', color='k')
```

Last 60 production rates of observations have been taken from model training data.

Using the above observation, oil production rates have been forecasted for upcoming 60 days.



x-axis – Years y-axis – Values Obtained

Figure 12. Data Visualization of Oil Production Forecasting

```
plt.figure(figsize=(15,5))





plt.scatter(x_train.index, y_train, label='Actual Oil Rates (Training)')
plt.plot(x_train.index, yp_train, color='k', label='Predicted Oil Rates (Training)')



plt.axvline(x_test.index[0], color='k')

# plt.scatter(X_test.index, y_test, color='green', label='Actual Oil Rates (Validation)')
plt.plot(x_test.index, yp_test, label='Predicted Oil Rates (Validation)', color='green')

plt.axvline(df_model_testing.index[0])

# plt.scatter(df_model_testing.index[:window], df_model_testing['BORE_OIL_VOL'][:window], label='Actual Future Rates')
plt.plot(df_model_testing.index[:window], next_forecasts, label='Model Forecasted Rates', color='brown')





plt.ylabel('Oil Production rates, STB/D', size=20);
plt.xlabel('Date', size=20)
plt.grid()
plt.legend()
```

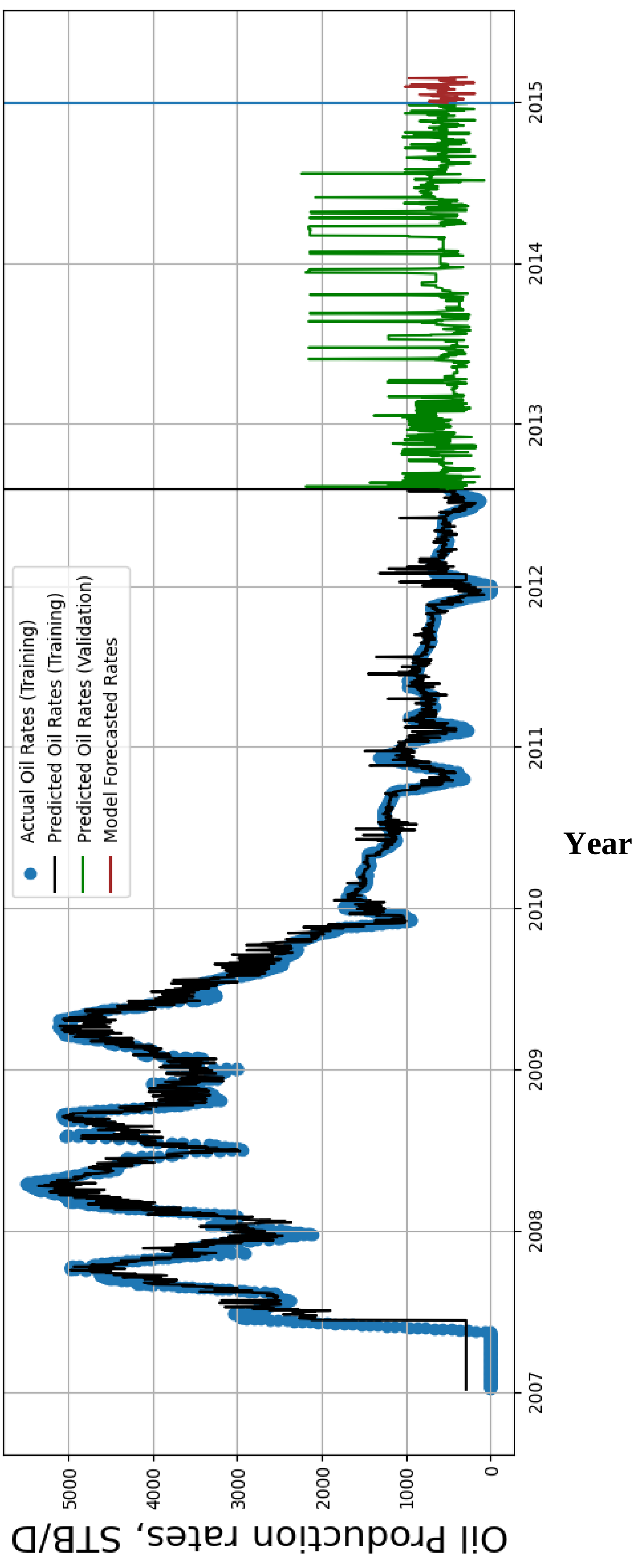


Figure 13. Data Visualization of Overall Performance of our model in training, testing and forecasting

9. Conclusion:

Machine learning can be an effective tool for oil production forecasting: By using machine learning algorithms, it is possible to create models that can accurately predict oil production based on historical data. This can be a valuable tool for oil companies and other stakeholders in the industry. Accuracy of the model depends on the quality of the data: The accuracy of the model will depend on the quality of the data used to train it. Inaccurate or incomplete data can lead to inaccurate predictions, so it is important to ensure that the data used is of high quality. Different machine learning algorithms may perform differently: Different machine learning algorithms may perform differently depending on the specific characteristics of the data. It is important to experiment with different algorithms to find the one that performs best for the specific task at hand.

10. References:

1) A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet -
Yanrui Ning, Hossein Kazemi, Pejman Tahmasebi

2) Forecasting the realized variance of oil-price returns using machine learning: Is there a role for U.S. state-level uncertainty? –
Oguzhan Cepni, Rangan Gupta, Daniel Pienaar, Christian Pierdzioch

3) Optimization of machine learning approaches for shale gas production forecast:
Muming Wang, Gang Hui, Yu Pang, Shuhua Wang, Shengnan Chen

4) Data-driven approach for hydrocarbon production forecasting using machine learning techniques:
Jaiyesh Chahar, Jayant Verma, Divyanshu Vyas, Mukul Goyal

5) Forecasting long-term world annual natural gas production by machine learning:
Doruk Sen, K. Irem Hamurcuoglu, Melisa Z. Ersoy, K.M. Murat Tunc,
M. Erdem Gunay

6) Production performance forecasting method based on multivariate time series and vector autoregressive machine learning model for waterflooding reservoirs:
Rui Zhang, Hu Jia

7) Multi-step-ahead crude oil price forecasting based on two-layer decomposition technique and extreme learning machine optimized by the particle swarm optimization algorithm:
Tingting Zhang, Zhenpeng Tang, Junchuan Wu, Xiaoxu Du, Kaijie Chen

8) Energy demand forecasting in seven sectors by an optimization model based on machine learning algorithms:
Majid Emami Javanmard, S.F. Ghaderi

9) Shale oil production prediction and fracturing optimization based on machine learning:
Chunhua Lu, Hanqiao Jiang, Jinlong Yang, Zhiqiang Wang, Miao Zhang,
Junjian Li

10) Forecasting multiphase flowing bottom-hole pressure of vertical oil wells using three machine learning techniques:
Nagham Amer Sami, Dhorgham Skban Ibrahim

11) Machine learning-based production forecast for shale gas in unconventional reservoirs via integration of geological and operational factors:
Gang Hui, Shengnan Chen, Yongming He, Hai Wang, Fei Gu

12) A Hybrid Model with Applying Machine Learning Algorithms and Optimization Model to Forecast Greenhouse Gas Emissions with Energy Market Data:
Majid Emami Javanmard, S.F. Ghaderi

13) Forecasting in a complex environment: Machine learning sales expectations in a stock flow consistent agent-based simulation model:
Ermanno Catullo, Mauro Gallegati, Alberto Russo

14) A systematic machine learning method for reservoir identification and production prediction:
Wei Liu, Zhangxin Chen, Yuan Hu, Liuyang Xu

15) Application of machine learning algorithms for predicting the engine characteristics of a wheat germ oil–Hydrogen fuelled dual fuel engine:
Kaliraj Shanmugaiah, Ankit Sonthalia, Yuvarajan Devarajan, Edwin Geo Varuvel

16) A novel decomposition ensemble model with extended extreme learning machine for crude oil price forecasting:
Lean Yu, Wei Dai, Ling Tang

17) Research on oilfield production forecasting technology under low-carbon background:
Yuxin Wang, Zexin Yang, Xiaoju Yin, Qingshan Zhu, Jianguo Lian

18) Machine learning techniques and data for stock market forecasting:
Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, Jari Porras

19) Machine Learning and Statistics: A Study for assessing innovative Demand Forecasting Models:
Nikolas Ulrich Moroff, Ersin Kurt, Josef Kamphues

20) Unsupervised machine learning technique for classifying production zones in unconventional reservoirs:
Y. Alhosin, Jebraeel Gholinezhad, Hesam Ghoochaninejad,Hossein Barati, James Buick, Paria Yousefi, Reham Alasar,Salam Al-Saegh