ConocoPhillips

Carnegie Mellon University

Project Report

Well Completion Optimization

1st August, 2024

Contributors:
Gautam Devadiga
Kristen Dmello
Roselle Hsu
Ryan Wang
Vignesh Sridhar

ConocoPhillips

Carnegie Mellon University

CONTENTS

Executive Summary

Introduction and Background

Client Background ..... Pg 6

Business Context - Introduction to Hydraulic Fracturing ..... Pg 6

Business Problem ..... Pg 6

Proposed Solution ..... Pg 7

Business Impact ..... Pg 7

Project Plan Workflow ..... Pg 8

Data

- Completion ..... Pg 9
  - Column & Definition
  - Cleaning Steps & Rationale
  - File Reference
- Production ..... Pg 16
  - Column & Definition
  - Cleaning Steps & Rationale
  - File Reference

Models

- Baseline Model: Linear Regression ..... Pg 18
  - Background Explanation
  - Result, Supporting Visuals & Interpretation
  - Code Reference
- Model 1: Random Forest ..... Pg 20
  - Background Explanation
  - Result, Supporting Visuals & Interpretation
  - Code Reference
- Model 2: XGBoost ..... Pg 25
  - Background Explanation
  - Result, Supporting Visuals & Interpretation
  - Code Reference
- Model Selection & Performance Comparison ..... Pg 35
  - Metrics used for comparison
  - Model strengths and drawbacks
  - Best Model Selection

Applications

Influential Parameters & Definition ..... Pg 37
Sensitivity Analysis ..... Pg 38
What-If Analysis ..... Pg 38
Code Reference ..... Pg 46
Future Integration Suggestion ..... Pg 47

Conclusions & Lessons Learned
Conclusions ..... Pg 47
Lessons Learned ..... Pg 48
Limitations of Current Framework ..... Pg 49
Quality Management Control ..... Pg 49
Budget ..... Pg 51

APPENDIX A: Glossary
APPENDIX B: Meeting Notes

Executive Summary

Client Introduction & Project Background

Over the summer of 2024, we collaborated with ConocoPhillips, a leading energy company primarily focused on the exploration and production of crude oil and natural gas. Our project aimed to enhance their current operations, specifically the "Well Completion" process, by leveraging advanced machine learning techniques to predict the final production amount and identify optimal operational settings, as well as suggesting beneficial adjustments for well completion that could increase the potential production of crude oil and natural gas.

Existing Solutions and Problems

While ConocoPhillips already utilizes machine learning models capable of predicting the final oil production amount based on various completion parameters, the completion process remains inherently complex, with numerous influencing factors creating uncertainties. The existing models forecast production based on historical data but lack the functionality to identify and prioritize the most critical factors due to the dynamic and interrelated nature of completion parameters, which involve areas such as geology, physics, chemistry, well design, hydraulic fracturing methods, and partnering crews. Furthermore, while the models can predict outcomes, they lack the extended capabilities needed to provide actionable insights for real-time operational adjustments during the completion process. This limitation means that if production forecasts are lower than expected, the models do not offer specific recommendations on how to tweak parameters, such as drilling techniques or fluid compositions, to enhance outcomes. Without additional analysis based on a well-performing model and further suggestions for adjustments, ConocoPhillips could miss opportunities to explore possibilities that could potentially increase production. Therefore, our solution involves conducting feature importance analysis, along with sensitivity and what-if analysis, to provide actionable insights and recommendations for optimizing operational parameters during the completion stage.

Our Solution

To address these challenges, we developed a couple of machine learning models using Linear Regression, Random Forest, and XGBoost, along with SHAP feature importance for important parameter analysis. We've selected the best-performing model based on whichever has the lowest MAPE ("Median" Absolute Percentage Error) and extended the application to include sensitivity and what-if analyses based on this best model. These further analyses are designed to help ConocoPhillips understand how to adjust variable settings during completion to achieve higher outcomes and to quantify the impact of adjustments on final production. By experimenting with different configurations such as selecting a couple of wells that represent the median with adjustments on a few of the most important features, our solution aims to identify optimal settings that could enhance final oil production.

Project Deliverables

- **Predictive Models**: Total of three models for predicting the combined production of oil and gas quantified in BOE (Barrels of Oil Equivalent)
  1. Linear Regression with Log Transformation (baseline model)
  2. Random Forest
  3. XGBoost (best-performing final model)

- **Code Repository**: All project codes written in python notebooks (with details regarding logic and methods) for the following sections will be housed in a GitHub repository with access provided to ConocoPhillips:
  1. Data Processing * 2 (1 for Completion, 1 for Production)
  2. Models *3 (Linear Regression Log Transformed, Random Forest, XGBoost)
  3. Further Analyses (e.g. feature importance, sensitivity analysis, what-if analysis)

- **Final Report**: A comprehensive report documenting the entire project in detail, major sections include:
  - Executive Summary: overall description for the project
  - Introduction and Background: e.g. challenges, proposed solutions, project methodologies, etc.
  - Data: processing rationale, with metadata for cleaned version
  - Models: the rationale behind model selection, assumptions made, testing and tuning method, etc.
  - Applications: sensitivity & what-if analysis, with interpretations of results

- **Project Artifacts**: Additional project artifacts, such as meeting notes, and project videos.

Configuration

For this project, we developed two Python notebooks for data processing - one for completion data and one for production data. These notebooks are designed to allow ConocoPhillips to feed in data that comes in the same format (aka. same column names, the same total number of columns) as initially provided, but minor modifications will be needed if future data dimensions expanded beyond the original version (e.g. adjust mapping dictionary, update separation row number for Data1 & Data2, introduce new features, etc.).

While codes for processing Completion Data require potential future adjustments, codes for the best-performing model (XGBoost) should run without modification if fed with data produced by the two data processing notebooks (Completion.ipynb & Production.ipynb). If there are new model settings to be tested (e.g. test size, trials of Optuna to be tested, etc.), this can be done by adjusting a few parameters in the “Configuration” code block.

As for separate Python notebooks for sensitivity and what-if analysis, they would not require further modifications when run on top of the XGBoost model, unless additional variables are introduced for further experimentation in the what-if analysis.

Currently, the final XGBoost model is designed as a standalone model without pre-configuration for future production. However, should needs emerge for deploying the model into production, a potential solution could be integrating the model into a scalable architecture using tools for containerization (e.g. Docker), for orchestration (e.g. Kubernetes), and a CI/CD pipeline to ensure continuous integration and deployment, or any other methods that are currently utilized by ConocoPhillips.

Future Steps

Due to the complicated nature of the completion process such as technical difficulties and limitations, it is very difficult and not fitting to prescribe a definitive set of completion parameters that will yield maximum production. Therefore, we opted for sensitivity analysis to showcase potential changes in production amount based on tweaks to a few critical parameters. The results from the what-if analysis will enable ConocoPhillips to identify which parameters to adjust and by how much, based on specific operational settings already in place, and some additional calculations for cost-effectiveness on intended adjustments vs. additional-expecting costs.

By implementing these solutions and following the outlined future steps, ConocoPhillips will be better equipped to optimize their completion processes, ultimately leading to more efficient and productive oil and gas extraction operations. Our approach not only addresses the immediate challenges but also provides an extendable framework for ongoing operational improvements.

Introduction and Background

Client Background

ConocoPhillips, a global energy company headquartered in Houston, Texas, was formed from the merger of Conoco Inc. and Phillips Petroleum Company on August 30, 2002. With operations spanning 13 countries, ConocoPhillips is a leading producer of crude oil and natural gas. In 2023, the company achieved a production rate of 1,826 thousand barrels of oil equivalent per day and maintained 6.8 billion barrels of oil equivalent in proved reserves. The company is committed to core values including safety, integrity, innovation, and teamwork, striving to meet the world's energy needs while ensuring sustainable and responsible practices.

Business Context - Introduction to Hydraulic Fracturing

Hydraulic fracturing, commonly known as fracking, is a critical technique used in the oil and gas industry to extract hydrocarbons from unconventional reservoirs such as shale formations. This process involves injecting a high-pressure fluid mixture, primarily composed of water and chemicals, into the wellbore to create fractures in the rock formation. The introduction of proppants, such as sand, into these fractures ensures they remain open, allowing oil and gas to flow more freely into the wellbore and ultimately to the surface. High-powered pumps are employed to inject the fluid and proppant mixture at the necessary pressure to initiate and sustain these fractures. Hydraulic fracturing has revolutionized the ability to tap into previously inaccessible reserves, significantly boosting production and transforming the energy landscape. However, it involves complex decision-making and precise control of various parameters to optimize production and maintain safety and environmental standards.

Business Problem

The hydraulic fracturing process, essential for extracting oil and gas, involves numerous complex and interrelated factors that significantly influence production outcomes. Currently, the decision-making process for setting completion parameters is manual, leading to inconsistencies and suboptimal production results. The primary challenge is to optimize these completion parameters to maximize oil and gas production, overcoming the limitations posed by manual decision-making and the complexity of the influencing factors. Specific challenges include:

1. Complex Decision-Making: The process involves numerous factors such as fracturing fluid volume, pressure, and wellbore geometry, each significantly influencing the outcome. The interaction between these parameters can be highly complex, making it difficult to predict the optimal settings.
2. Manual Process: The manual decision-making process can lead to inconsistencies and suboptimal production results. Operators must rely on their experience and judgment, which can vary significantly, resulting in inconsistent outcomes.

Proposed Solution

To address these challenges, we developed a comprehensive solution involving three key components:

1) Predictive Model

Develop a predictive model to estimate the final production amount of oil and gas combined, measured in Barrels of Oil Equivalent (BOE), given specific settings of well completion parameters. This model utilizes XGBoost to analyze historical data and predict BOE production.

2) Sensitivity Analysis

Conduct sensitivity analysis to understand the impact of various well completion parameters on production outcomes. This analysis will help identify which parameters have the most significant influence on production, allowing for more focused optimization efforts.

3) What-if Scenarios

Develop a framework for what-if scenario analysis to explore the potential outcomes of different well completion strategies. This framework will allow users to simulate various parameter settings and assess their impact on production. By evaluating multiple scenarios, operators can make more informed decisions and select the most effective strategies for maximizing oil and gas production. This tool will also help in planning and adjusting strategies in response to changing conditions and objectives.

Business Impact:

By utilizing our predictive model to optimize the design parameters of wells and accurately predict production levels, our project will deliver substantial value to ConocoPhillips. The business impact is multifaceted, focusing on both production enhancement and cost efficiency:

1. **Enhanced Production of BOE (Barrels of Oil Equivalent):**

   The implementation of our model is expected to significantly increase the production of oil and gas, measured in BOE. Through precise adjustment of well completion parameters based on data-driven insights, ConocoPhillips can achieve higher extraction rates and maximize the output from each well.

2. **Significant Cost Savings:**

   Optimizing completion parameters not only boosts production but also leads to considerable cost savings. By reducing inefficiencies, the project can save ConocoPhillips an estimated $10 to $20 million.

Project Plan Workflow

Over the course of 12 weeks, we approached the project, ensuring comprehensive data analysis, model development, sensitivity analysis, and what-if scenario evaluation. Our project plan below will provide a visual representation of our workflow, highlighting key milestones, deliverables, and the critical path to project completion.

First Half of the Project

This image covers the initial phases of the project, including group acquaintance, data collection, project scoping, exploratory data analysis (EDA), data preprocessing, and base model building:

Major Task Phase & Subtasks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
							Midterm
	5/13 ~ 5/19	5/20 ~ 5/26	5/27 ~ 6/2	6/3 ~ 6/9	6/10 ~ 6/16	6/17 ~ 6/23	6/24 ~ 6/30
Group Acquaintance & Collect Data							
Internal kickoff, advisor kickoff, client kickoff							
Advisor kickoff							
Industry background study							
Collect data from client							
Project Scope & Brief EDA							
Study data given + preliminary EDA							
Develop critical thinking exercise + project scope							
Data Preprosessing, Cleaning, Feature Engineering							
Data cleaning & feature engineering - Completion Data							
Data cleaning & feature engineering - Production Data							
Build Base Models							
Baseline model (log regression)							
Random Forest							
Prepare Midterm Presentation							

Second Half of the Project

This image outlines the latter phases, including model refinement, additional feature engineering, sensitivity analysis, what-if scenarios, and the preparation for the final presentation:

Major Task Phase & Subtasks	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
		Midterm					Final
	6/17 ~ 6/23	6/24 ~ 6/30	7/1 ~ 7/7	7/8 ~ 7/14	7/15 ~ 7/21	7/22 ~ 7/28	7/29 ~ 8/2
Midterm Presentation		**6/26 Wed.**					
Modify, Refine & Construct							
Additional feature engineering - Completion							
Additional feature engineering - Production							
Refine Random Forest Model (IP180 vs. IP365) In progress, discussed with cli							
Create XGBoost Models (IP180 vs. IP365)							
SHAP feature importance							
Aggregate Solutions							
Create metadata for Completion & Production							
Select best performing model & define important features							
Create supporting visuals (residual plot, predicted vs. actual)							
Sensitivity analysis							
What-if analysis							
Final Presentation							
Structure layout							
Prepare final presentation content + adjusment & alignment							
Team Rehearsal (in person)							

Data

I. Completion Dataset

File Reference

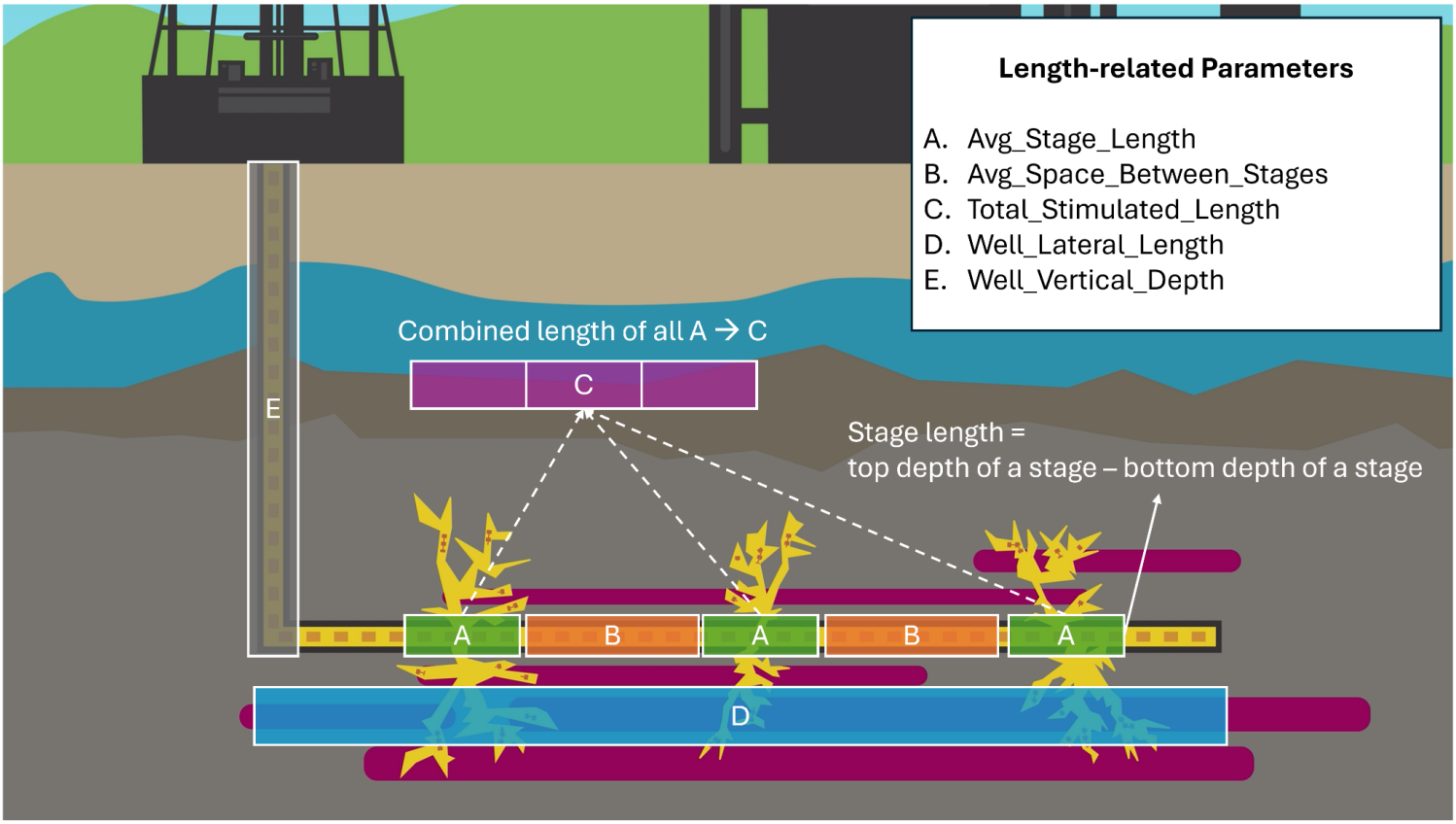
- Cleaning Code: Completion.ipynb
- Cleaned Data: Completion.xlsx
- Metadata: Metadata_Completion+Production.xlsx

Column & Definition

*** For data type & constraints, refer metadata

Column Name	Definition
UWI	Unique Well ID
Total_Stages	number of stages for that Well (take total count, not the max. of stage_num from data)
Avg_FracDays_per_Stage	Avg. number of days to frac a stage for that well
Avg_Breakdown_Pressure	Avg. breakdown pressure of all stages of a well
AVG_PROPPANT_PER_STAGE	Avg. amount of proppant used per stage of a well (excluding stage1 because stage1 tends to fluctuate)
Total_Proppant	Total proppant of all stages of a well (including stage1)
STDEV_PROPPANT_PER_STAGE	Standard deviation of proppant used per stage (excluding stage1)
AVG_STIM_TREAT_RATE	Avg. rate of how fast the hydraulic fracturing fluid is injected into the reservoir formation layer
STDEV_STIM_TREAT_RATE	Standard deviation of the rate of how fast the hydraulic fracturing fluid is injected into the reservoir formation layer
Total_Stimulated_Length	Total length of all stimulated stages of a well (will be smaller than Well_Lateral_Length) in feet
Well_Lateral_Length	Total lateral length of a well (total stage length + total spacing) in feet
Well_Vertical_Depth	Total vertical length of a well (aka. true vertical depth) in feet
Avg_Stage_Length	Avg. length of a stage (of all stages of a well) in feet
Avg_Space_Between_Stages	Avg. length of space between stages (of the same well)
Total_Clean_Volume_Pumped	Total clean volume pumped for a well
AVG_CLEAN_VOLUME_PUMPED_perStage	Avg. volume of clean fluid pumped per stage (of a well)
AVG_SLURRY_VOLUME_PUMPED_perStage	Avg. volume of slurry pumped per stage (of a well)
AVG_FLUID_VISCOCITY	Avg. fluid viscosity (aka. PPG, proppant per stage)
STIM_INT_TREAT_TYPE_Hyd Frac-Gelledwater	Whether the stimulation method is hydraulic fracturing with "Gelledwater"
STIM_INT_TREAT_TYPE_Hyd Frac-Other	Whether the stimulation method is hydraulic fracturing within "Other" category
STIM_INT_TREAT_TYPE_Hyd Frac-Slickwater	Whether the stimulation method is hydraulic fracturing with "Slickwater"
STIM_INT_TREAT_TYPE_Hyd Frac-Zipper	Whether the stimulation method is hydraulic fracturing with "Zipper"
STIM_INT_TREAT_TYPE_Hydraulic Fracture	Whether the stimulation method is "General" hydraulic fracturing
STIM_INT_TREAT_TYPE_Sand Frac	Whether the stimulation method is "Sand" fracturing
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000002	Whether the stimulation process is carried out by companyID "2019000002"
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000003	Whether the stimulation process is carried out by companyID "2019000003"
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000004	Whether the stimulation process is carried out by companyID "2019000004"
STIM_COMPANY_STIM_TREAT_COMPANY -	Whether the stimulation process is carried out by companyID

2019000005	"2019000005"
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000007	Whether the stimulation process is carried out by companyID "2019000007"
STIM_COMPANY_STIM_TREAT_COMPANY - 2019110002	Whether the stimulation process is carried out by companyID "2019110002"



Cleaning Rationale

Dimension

- Before cleaning: (20463, 182)
- After cleaning: (458, 30)
- Total unique wells after cleaning: 458

Data Issues Before Cleaning

1. **Multiple Data Sources**: Current data is a combination of 2 data sources (call them Data1 & Data2), with different numbers of total columns, different column names, and slightly different unique values within a few columns. Data1 starts from row2 ~ row2471, Data2 starts from row2472 and onwards.

2. **Missing Data**: Some important variables we intended to keep don't appear in both Data1 and Data2, e.g. RIG_CONTRACTOR, PLANNED_FORMATION, L48DW_DISTRICT, L48DW_AREA, so we'll need to remove those that don't exist in both data sources since they cannot be imputed.

Steps for Transformation

1. Create a mapping dictionary for Data1 & Data2

- Current Dictionary for selected columns:

Data1	Data2
PRIMARY_JOB_TYPE	PROJECT_TYPE
STIM_COMPANY	STIM_TREAT_COMPANY
BREAKDOWN_PRESS	BREAKDOWN_PRESSURE
STIM_INT_PROPPANT_TOTAL	PROPPANT_IN_FORMATION
STIMULATION_TREAT_TYPE	TREAT_AVG_RATE
STIM_START_DATE	STG_START_DATE
STIM_END_DATE	STG_END_DATE
STIM_INT_TOP_DEPTH	STG_TOP_DEPTH
STIM_INT_BTM_DEPTH	STG_BOTTOM_DEPTH
STIM_INT_CLEAN_VOLUME_PUMPED	STG_CLEAN_VOLUME_PUMPED
STIM_INT_SLURRY_VOLUME_PUMPED	STG_SLURRY_VOLUME_PUMPED
STIM_INT_TOP_DEPTH_TVD	TOP_DEPTH_TVD
STIM_INT_STAGE_NUMBER	STG_NUMBER

Note: this dictionary will need to be manually adjusted if any of the below conditions satisfy:

- Any of the original columns in Data1 or Data2 changed name in the future
- New columns are introduced in both Data1 & Data2 that could be helpful in prediction

2. Slicing data into Data1 & Data2 based on row number

- Current separation row number is Data1: row1 ~ row2741; Data2: row2742 onwards

Note: this row number is currently hard-coded and will need to be manually adjusted in the future should the original dimension for Data1 extended beyond row2741

3. Keep only relevant records

- Apply 2 initial filters:
  (1) STIM_INT_TREAT_TYPE: keep records that contain "frac", and exclude rows with missing value
  (2) PRIMARY_JOB_TYPE: remove all records that said "RECOMPLETION" as these records aren't original and could be modified or tampered with, for blank records, we're assuming they are original completion records thus not removing them
- Remove any columns that have all NULL value

4. Feature Engineering

- **Impute Missing Values with Median Within Wells**: for missing numeric values within each well, replace NULLs with the median value of the corresponding column for that specific well
- **Set a minimum for Proppant** (STIM_INT_PROPPANT_TOTAL): set a minimum threshold of 30,000, anything below this should be adjusted based on imputation. Use the median proppant amount from the same stage number of other wells to fill in the missing proppant for a particular stage.

  E.g. If Well A's proppant for stage3 is below 30000, find the median of all stage3's proppant, and fill in for WellA's stage3 proppant amount. This method applies to all other NULL values for proppant as well

- **Replace Zero Values with Median Within Wells**: for columns that should not logically contain zero values, replace zeros with the median value of the corresponding column within the same UWI group
- **Impute Out-of-Range Values with Median Within Wells**: for values outside a specified range in a specific column, replace them with the median value of that column within the same UWI group.

  E.g. STIM_INT_TREAT_AVG_RATE should be between 50 & 150

5. Feature Engineering

- Format all date columns into date format
- Add some new numerical columns that are based on existing numerical columns, as well as transforming categorical variables into dummies
- Total new variables generated during feature engineering:
  - Numerical: 17
  - Categorical: 12
- **Note**: if there are new features introduced in both Data1 & Data2 that are deemed influential, manual creation for new variables during feature engineering stage will be required

Numerical Variables

*** Orange_texts mean column name from original data

New Variable Name	Aggregation Method	Interpretation & Calculation Method
Total_Stages	Count	Count of all stage numbers for a well as the total stages. Not taking the max because it's possible for a well to have missing records for some stages, e.g. a well could have stages 1,3,5,6, if taking 6 as total stages it'd be wrong, here it will count as 4. *** We also used Total_Stages to filter out any wells that have fewer than 3 stages since it's not practically reasonable.

Avg_FracDays_per_Stage	Average	Avg number of days for a crew to frack a stage. This could imply the efficiency of the crew. = (stim_end_dates – stim_start_dates) / Total_Stages = total days spent for fracking / total number of stages.
Avg_Breakdown_Pressure	Average	Average BREAKDOWN_PRESS by UWI
Total_Proppant	Sum	Total proppant of a well. = sum of STIM_INT_PROPPANT_TOTAL = sum of proppant from each stage of a well
AVG_PROPPANT_PER_STAGE	Average	Average proppant used per stage (STIM_INT_PROPPANT_TOTAL) within a well, excluding stage1 proppant (stage1 usually has the lowest proppant and is more volatile)
STDEV_PROPPANT_PER_STAGE	Standard Deviation	SD of proppant used per stage (STIM_INT_PROPPANT_TOTAL) excluding stage1's proppant
AVG_STIM_TREAT_RATE	Average	Average of treatment rate of a stage (STIM_INT_TREAT_AVG_RATE) within a well
STDEV_STIM_TREAT_RATE	Standard Deviation	SD of treatment rate of a stage (STIM_INT_TREAT_AVG_RATE) within a well
Total_Stimulated_Length	Sum	Total length of all stage lengths within a well = sum of all (STIM_INT_BTM_DEPTH – STIM_INT_TOP_DEPTH)
Well_Lateral_Length	Sum	Total lateral length of an entire well (lengths of stages + all spacings in between) = bottom depth of max stage number (STIM_INT_BTM_DEPTH) – top depth of min stage number (STIM_INT_TOP_DEPTH)
Well_Vertical_Depth	Average	= average of true vertical depth of all stages within a well (taking avg because the TVD for all stages are very similar, with only slight difference) = average (STIM_INT_TOP_DEPTH_TVD)

Avg_Stage_Length	Average	= average of all stage lengths = average of all (STIM_INT_BTM_DEPTH – STIM_INT_TOP_DEPTH)
Avg_Space_Between_Stages	Average	Average of all spacings between stages = (Well_Lateral_Length – Well_Lateral_Length * Total_Stages) / (Total_Stages - 1)
Total_Clean_Volume_Pumped	Sum	Sum of all clean volume pumped for all stages within a well = sum of STIM_INT_CLEAN_VOLUME_PUMPED
AVG_CLEAN_VOLUME_PUMPED_perStage	Average	Average of all clean volume pumped for all stages within a well = average of STIM_INT_CLEAN_VOLUME_PUMPED
AVG_SLURRY_VOLUME_PUMPED_perStage	Average	Average of all slurry pumped for all stages within a well = average of STIM_INT_SLURRY_VOLUME_PUMPED
AVG_FLUID_VISCOCITY	Average	Average of all fluid viscosity for all stages within a well, aka. PPG. = average of [(STIM_INT_PROPPANT_TOTAL – STIM_INT_CLEAN_VOLUME_PUMPED) / 42]

Categorical Variables

- 2 meaningful categorical variables we're able to keep and mapped from both Data1 & Data2:
  (1) STIM_INT_TREAT_TYPE, breaking down into:
    - STIM_INT_TREAT_TYPE_Hyd Frac-Gelledwater
    - STIM_INT_TREAT_TYPE_Hyd Frac-Other
    - STIM_INT_TREAT_TYPE_Hyd Frac-Slickwater
    - STIM_INT_TREAT_TYPE_Hyd Frac-Zipper
    - STIM_INT_TREAT_TYPE_Hydraulic Fracture
    - STIM_INT_TREAT_TYPE_Sand Frac

  (2) STIM_COMPANY, breaking down into:
    - STIM_COMPANY_STIM_TREAT_COMPANY - 2019000002
    - STIM_COMPANY_STIM_TREAT_COMPANY - 2019000003
    - STIM_COMPANY_STIM_TREAT_COMPANY - 2019000004
    - STIM_COMPANY_STIM_TREAT_COMPANY - 2019000005
    - STIM_COMPANY_STIM_TREAT_COMPANY - 2019000007
    - STIM_COMPANY_STIM_TREAT_COMPANY - 2019110002

6. **Remove columns that aren't aggregated at Well level**
- Remove the columns that we initially kept and used for transformation during feature engineering

7. **Aggregate data by Well level**

II. Production Data

File Reference

- Cleaning and Aggregation Code: Production_aggregation.ipynb
- Cleaned and Aggregated Data: aggregated_data_ip180.xlsx
- Metadata: Metadata_Completion+Production.xlsx

Column & Definition

*** For data type & constraints, refer metadata

UWI	Unique Well ID
CYCLE_YEARS	List of years that the well have production records
CYCLE_MONTHS	List of months that the well have production records
OIL_GAS_CODE	Well is producing gas or oil
LEASE_OIL_PROD_VOL	Total amount of oil produced in Barrels
LEASE_GAS_PROD_VOL	Total amount of gas produced in cubic feet
GAS_PROD_BOE	Total amount of gas produced (in BOE)
BOE	Combined amount of production from gas & oil (in BOE)

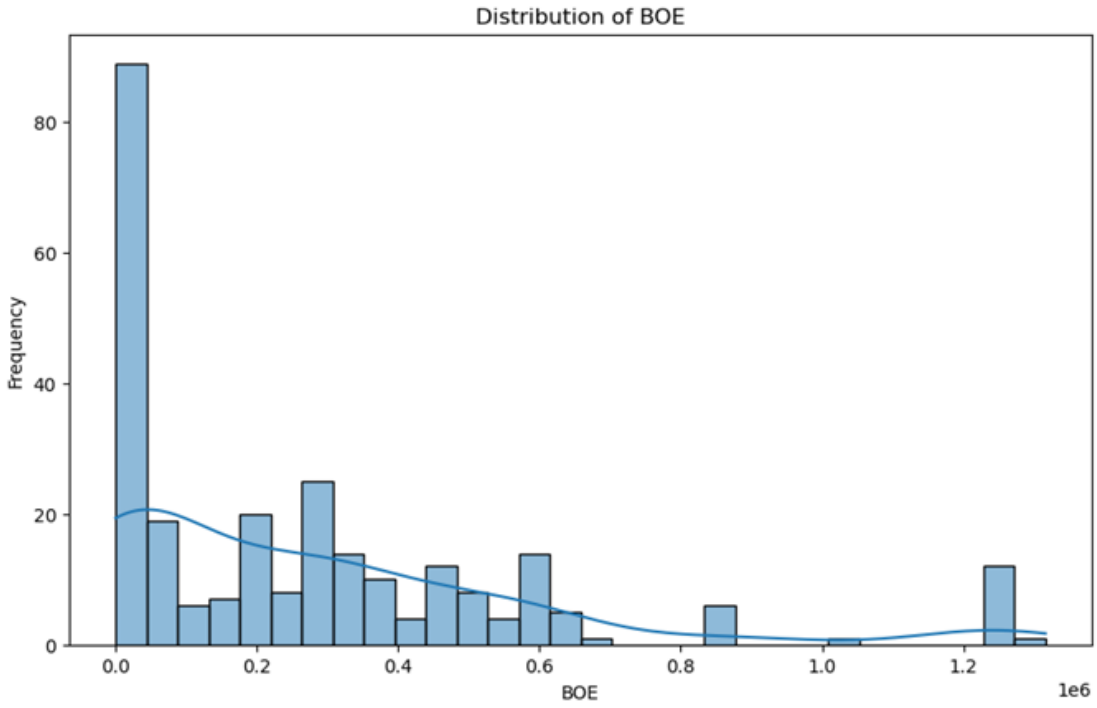
The production data comprises 21,968 records, detailing the oil and gas production volumes for each UWI, categorized by cycle year, month and oil gas code.

Steps for Cleaning and Aggregation:

1. **Remove columns that are not useful**
   Columns LEASE_GAS_LIFT_INJ_VOL and LEASE_COND_PROD_VOL were dropped.
2. **Convert production volumes to BOE (Barrels of Oil Equivalent)**
   Oil is measured in barrels whereas gas is measured in cubic feet, so BOE is a measure that helps to standardize and compare the energy content of oil and natural gas, on a common basis. One barrel of oil is deemed to have the same amount of energy content as 6,000 cubic feet of natural gas, hence we divide the gas production volume by 6000 and then Combine the oil and gas production volumes to get BOE.
3. **Remove 0 BOE volumes records**
   All 0 BOE records are removed from the dataset before aggregation.
4. **Aggregate BOE at UWI level**
   In our model, we analyze two datasets: for IP180, we aggregate the BOE from the first 6 months of production,

and for IP360, we aggregate the BOE from the first 12 months, all at the well level using the unique well identifier (UWI). After aggregation we get 324 unique wells for IP180 both 360.

Distribution



The distribution of BOE after removing all 0 values is highly right-skewed, meaning most values are concentrated at the lower end of the scale, with a long tail extending towards higher values. This skewness can negatively impact the performance of a machine learning model, so crucial to account for this skewness to ensure accurate and reliable predictions.

Models

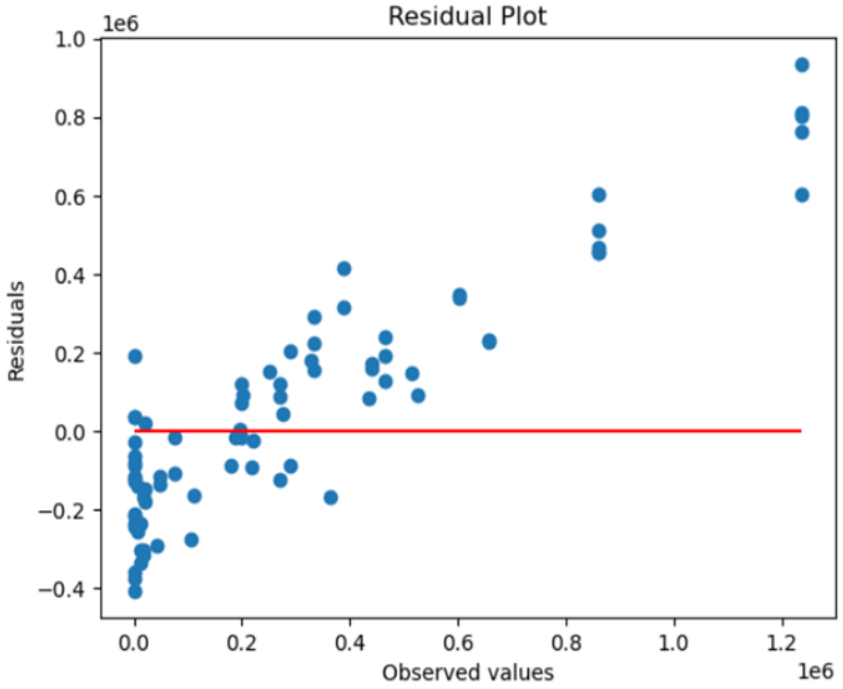
1. Linear Regression

Model Background

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. The goal is to find the best-fitting straight line (the regression line) that minimizes the differences (residuals) between observed values and the values predicted by the model. Linear regression assumes a linear relationship between the dependent and independent variables and is widely used due to its simplicity and interpretability.

Regression Steps:

1. **Residual Analysis:** Before performing linear regression, it is crucial to check if the relationship between the predictors and the response variable is linear. The main reason for conducting residual analysis is to verify that the assumptions of the regression model are met. These assumptions include the linearity of the relationship between predictors and the response variable and the constancy of the variance of the residuals across all levels of the fitted values.
   Typically, for linear data, the residuals should be randomly scattered around the horizontal axis (y=0), with no obvious patterns, trends, or systematic structures.



However, looking at our plot, the residuals form a pattern where the residual values increase with the observed values. This suggests that some transformations are necessary to stabilize the variance.

2. **Scale:** We scale continuous features to ensure that all features contribute equally to the model, reducing the dominance of features with larger values.
3. **Multicollinearity:** To reduce redundancy, we dropped features that show multicollinearity. For this, we use the Variance Inflation Factor (VIF), which quantifies how much the variance of a particular predictor's coefficient is inflated because of correlations with other predictors.
4. **Log Transform:** We apply a log transformation to the target variable to stabilize its variance, reduce skewness, and make its distribution more normal. This improves the model's fit and helps meet the assumptions of linear regression. Log transformation also reduces the impact of outliers by compressing extreme values, mitigating their influence on the model.
5. **Fit:** We iteratively refit the model, removing features with p-values greater than 0.05 to ensure that only statistically significant predictors are included. This helps refine the model for better accuracy and reliability.

Model Result

OLS Regression Results

Dep. Variable:	BOE	R-squared:	0.583
Model:	OLS	Adj. R-squared:	0.571
Method:	Least Squares	F-statistic:	50.08
Date:	Tue, 25 Jun 2024	Prob (F-statistic):	2.87e-32
Time:	19:35:27	Log-Likelihood:	-394.97
No. Observations:	185	AIC:	801.9
Df Residuals:	179	BIC:	821.3
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	11.8908	0.195	61.052	0.000	11.506	12.275
Avg_Breakdown_Pressure	0.9206	0.248	3.717	0.000	0.432	1.409
Well_Vertical_Depth	-1.0142	0.258	-3.925	0.000	-1.524	-0.504
AVG_FLUID_VISCOCITY	0.4068	0.155	2.629	0.009	0.102	0.712
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000003	2.7664	0.631	4.384	0.000	1.521	4.012
OIL_GAS_CODE_G	-5.1068	0.409	-12.484	0.000	-5.914	-4.300

Omnibus:	3.571	Durbin-Watson:	2.161
Prob(Omnibus):	0.168	Jarque-Bera (JB):	4.238
Skew:	0.016	Prob(JB):	0.120
Kurtosis:	3.741	Cond. No.	5.71

The regression model includes the following predictors and their corresponding coefficients:

- Avg_Breakdown_Pressure: 0.9206 (p = 0.000)
- Well_Vertical_Depth: -1.0142 (p = 0.000)
- AVG_FLUID_VISCOSITY: 0.4068 (p = 0.009)

- STIM_COMPANY_STIM_TREAT_COMPANY - 201900003: 2.7644 (p = 0.000)
- OIL_GAS_CODE_G: -5.1068 (p = 0.000)

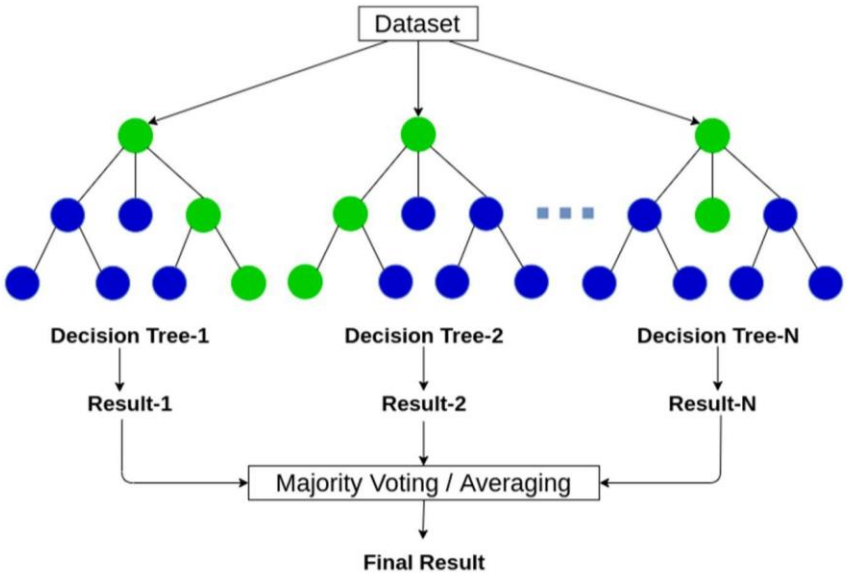
The significant predictors suggest a strong relationship with BOE production, with positive coefficients indicating a direct relationship and negative coefficients indicating an inverse relationship.

However, it is important to note that some key predictors such as lateral length, total stages, total proppant, and clean volume pumped were not selected as significant features in this model. Based on physical expectations, these factors are known to significantly impact BOE production. This suggests that the model may not fully capture all the factors affecting production, potentially limiting its accuracy and completeness.

2. Random Forest

Model Background

Random Forest is an ensemble learning method primarily used for classification and regression tasks. The key idea behind Random Forest is to combine multiple decision trees to form a forest and improve the model's overall performance and robustness.



How Random Forest Works

1. **Bootstrapping (Bagging)**: Random Forest uses a technique called bootstrap aggregating, or bagging, to generate multiple datasets from the original data. Each new dataset is created by randomly sampling the original data with replacement. This means some data points may appear multiple times in a new dataset, while others may not appear at all.
2. **Building Multiple Decision Trees**: For each bootstrapped dataset, a decision tree is constructed. These trees are trained independently and fully grown without pruning. Random Forest introduces randomness in the construction of each tree by selecting a random subset of features to split on at each node, rather than considering all features. This helps to create a diverse set of trees and reduces the correlation between them.
3. **Aggregation**: Once all the trees are built, the Random Forest model aggregates their predictions. For classification tasks, the final prediction is made based on the majority vote of the individual trees. For regression tasks, the final prediction is the average of all the trees' outputs.

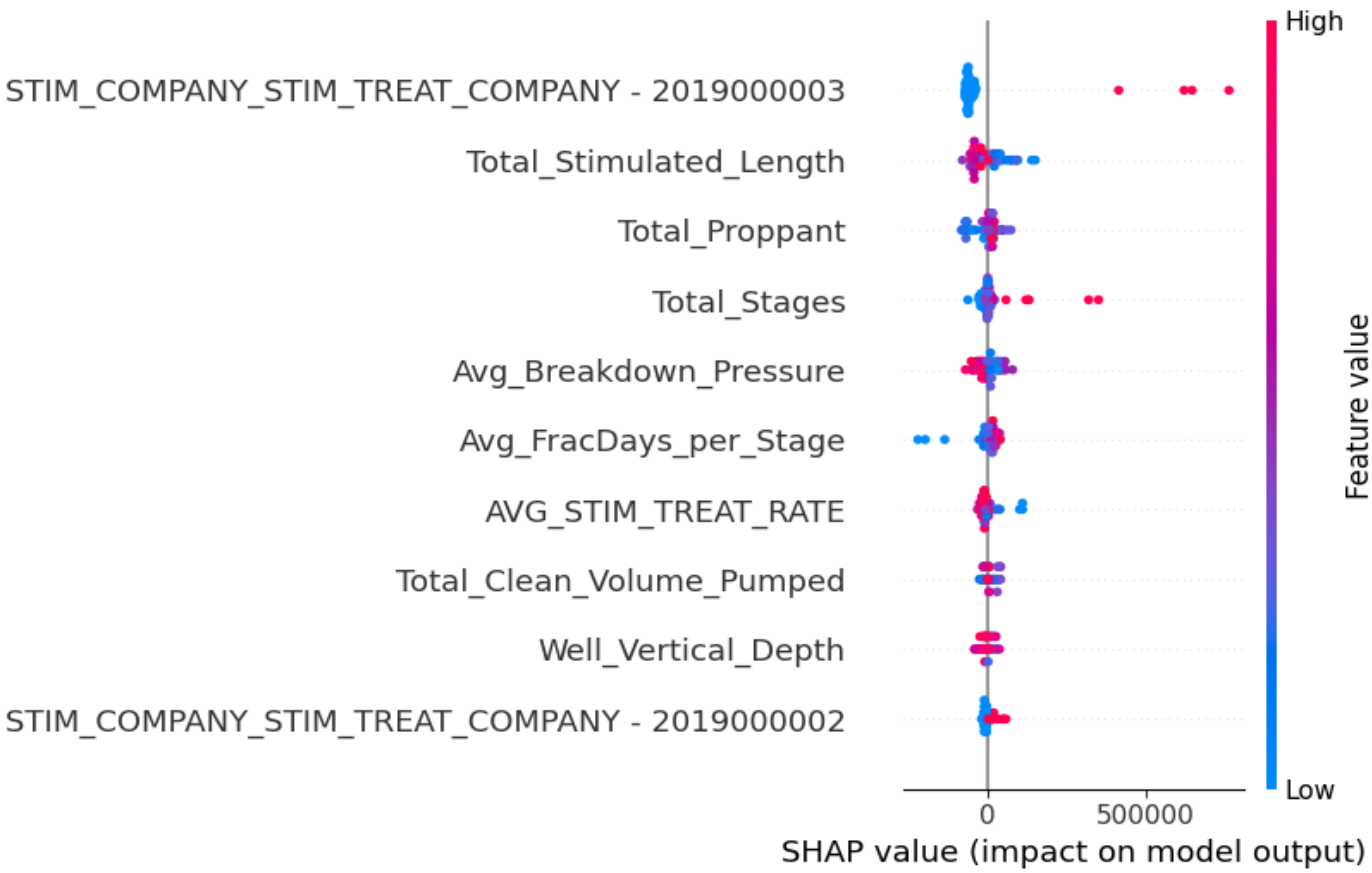
SHAP Analysis to pick the best features:

SHAP is a method used to explain the output of machine learning models. It provides a way to interpret the predictions by assigning each feature an importance value for a particular prediction.

How Does SHAP Work?

- SHAP values are based on game theory, specifically the Shapley value.
- Each feature’s contribution to a prediction is calculated by considering all possible combinations of features.
- The SHAP value indicates both the direction (positive or negative impact) and the magnitude (how much it influences) of the feature on the prediction.

Top 10 features in order of importance:



These features are then used to train the model to have the best prediction outcome.

Hyperparameter Tuning

Hyperparameter tuning of XGBoost was conducted using Optuna, a powerful framework designed for multiple machine learning models. Below are the hyperparameters we tuned for this model and their ranges.

Key Hyperparameters Tuned

1. **n_estimators**: The number of trees in the forest.
2. **max_depth**: The maximum depth of each tree.
3. **min_samples_split**: The minimum number of samples required to split an internal node.
4. **min_samples_leaf**: The minimum number of samples required to be at a leaf node.

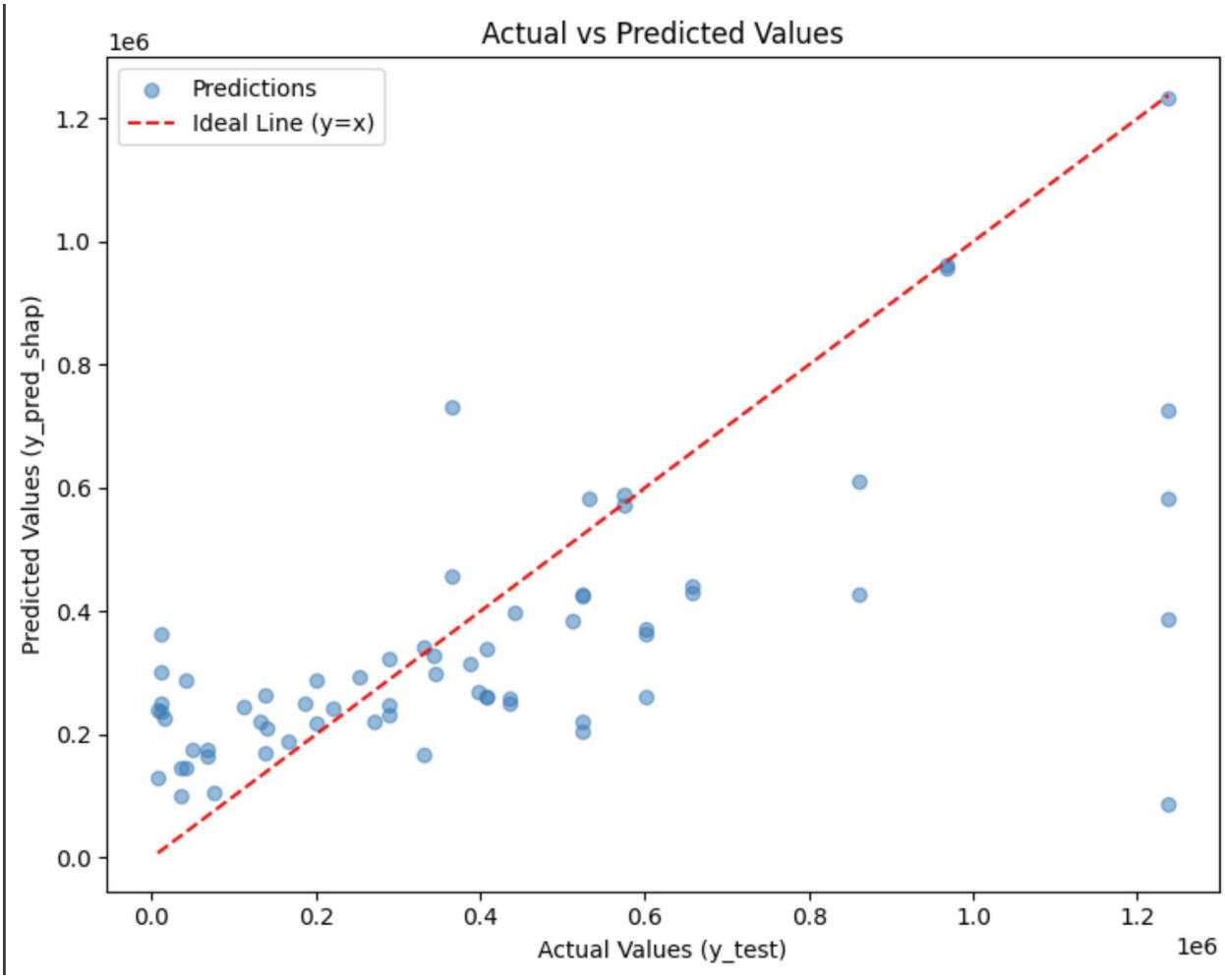
Hyperparameter Tuning Process with Optuna

1. **Objective Function**:
    - The `objective` function is defined to encapsulate the model training and evaluation process.
    - Within this function, the hyperparameters `n_estimators`, `max_depth`, `min_samples_split`, and `min_samples_leaf` are suggested using Optuna's `trial.suggest_int` method.
    - A Random Forest Regressor is initialized with these hyperparameters and trained on the training dataset (`X_train_top_10` and `y_train`).
    - The model's performance is evaluated on the test dataset (`X_test_top_10` and `y_test`) which is then returned as the objective value to be maximized.
2. **Optimization:**
    - An Optuna study object is created with the direction set to 'maximize'.
    - The study's optimize method is called with the objective function and the number of trials (n_trials=100), meaning the objective function will be evaluated 100 times with different hyperparameter values.
3. **Best Parameters and Score:**
    - After the optimization process, the best set of hyperparameters and the corresponding best R^2 score are extracted from the study object.
4. **Model Training with Best Parameters:**
    - The Random Forest Regressor is re-initialized with the best hyperparameters obtained from the tuning process and trained on the entire training dataset.
    - The model's performance is evaluated on both the test and training datasets using various metrics: R^2 score, Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Median MAPE.

Model Performance

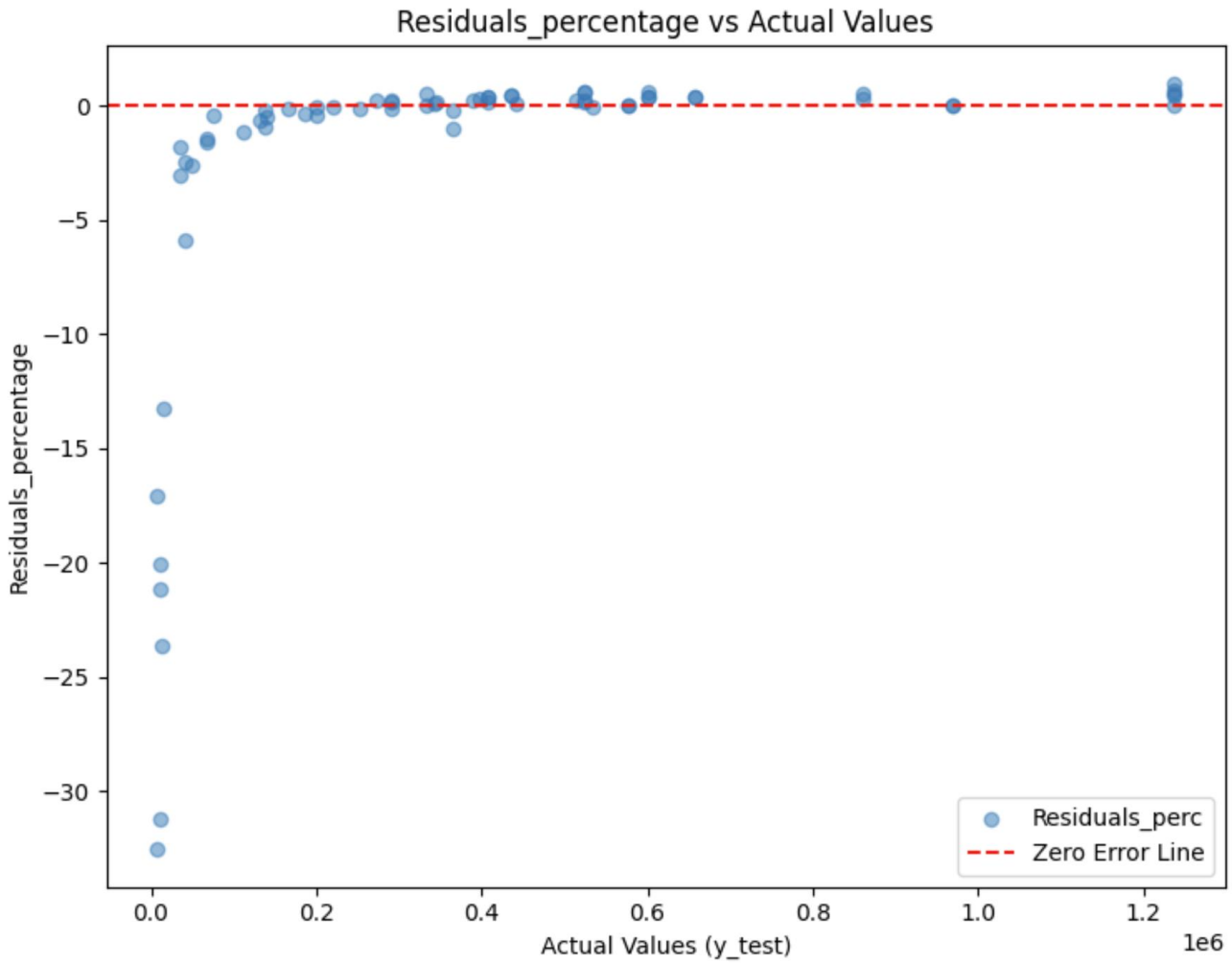
Test set: Median Mean Absolute Percentage Error: 39%

Prediction Analysis



Predicted values are plotted against actual values to compare with the x=y trendline that represents perfect prediction. Based on the plot, data points are scattered around the trendline and a few of them are away from the trendline, showing good prediction performance but it can still be improved.

While the Random Forest Regressor model makes generally accurate predictions, as indicated by the clustering around the ideal line, there is room for improvement. The presence of outliers and scattered points suggests potential areas for model refinement.



Residuals percentage is plotted against actual values. For low actual value wells (low BOE), the residual percentage was exceptionally higher than wells with larger BOE. The reason is low production level wells have similar completion parameters with high production level wells, but they are the minority of the dataset, making the model incapable of predicting accurately for those wells.

3. XGBoost

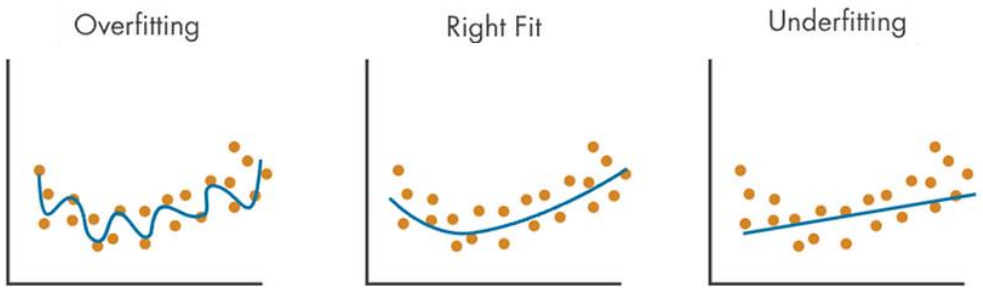
Model Background

XGBoost stands for “Extreme Gradient Boosting”. It is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate way.

XGBoost is used for supervised learning problems, where we use the training data x_i (with multiple features) to predict a target variable y_i. XGBoost can perform both classification and regression tasks. Within the scenario of this project, we used XGBoost to do regression to predict the BOE production level of hydraulic fracturing wells.

Why choose this model?

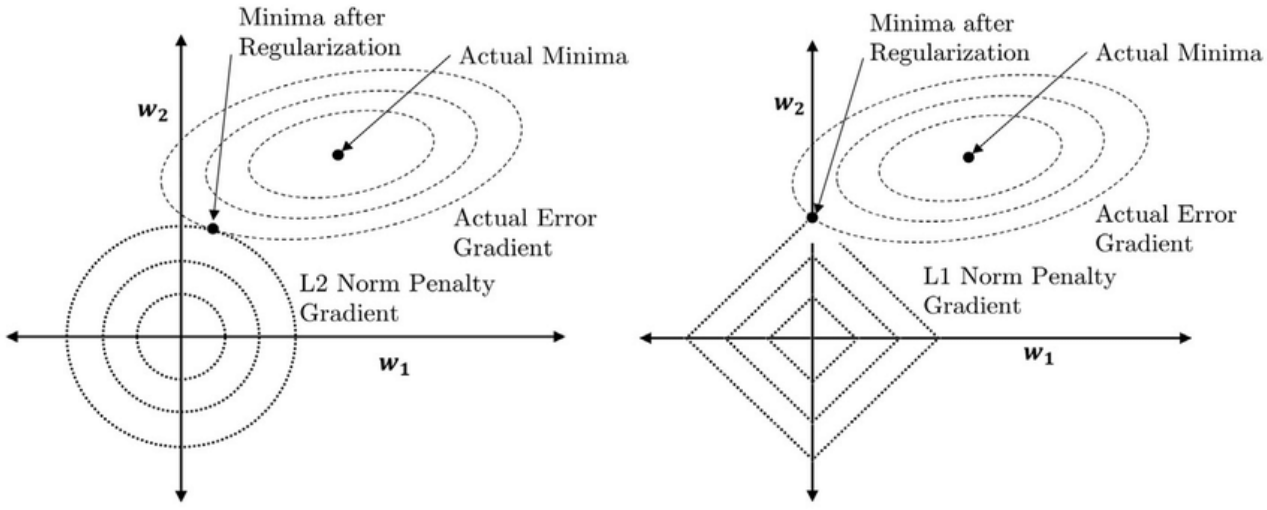
As previously mentioned in the 'Data' section, we have only 242 data points for model training. Limited training data often leads to model overfitting, where the model becomes too complex and captures noise in the dataset, resulting in poor performance on unseen data.



This situation makes XGBoost an optimal choice due to its exceptional ability to handle limited training data effectively. There are several reasons.

1. Regularization

Regularization helps to prevent overfitting by penalizing more complex models, thus encouraging simpler models that generalize better to unseen data. By incorporating both L1 (Lasso) and L2 (Ridge) regularization, XGBoost balances the complexity of the model with its performance on the training data.



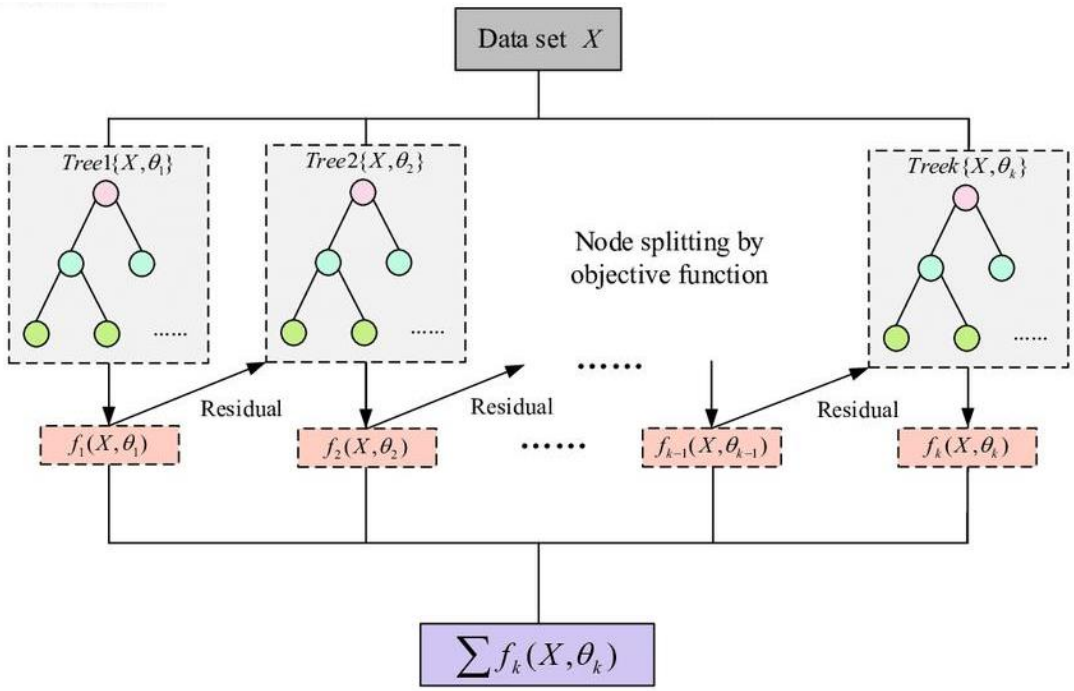
L1 regularization (right) adds the **absolute value** of the magnitude of coefficients as a penalty term to the loss function. This encourages sparsity in the model parameters, effectively driving some coefficients to zero and thus selecting a simpler model.

L2 regularization (left) penalizes the loss function by adding the **squared value** of the magnitude of coefficients. This discourages large coefficients by shrinking them towards zero but does not force them to be exactly zero, resulting in a smoother model with small, distributed weights.

2. Boosting Ensemble

One of the main reasons why XGBoost is so powerful is because it uses an ensemble technique called boosting. An ensemble technique in machine learning is a method that combines multiple individual models to produce a more robust and accurate predictive model. By aggregating the predictions of several models, ensemble methods aim to reduce errors, improve performance, and enhance the stability of the predictions compared to using a single model.

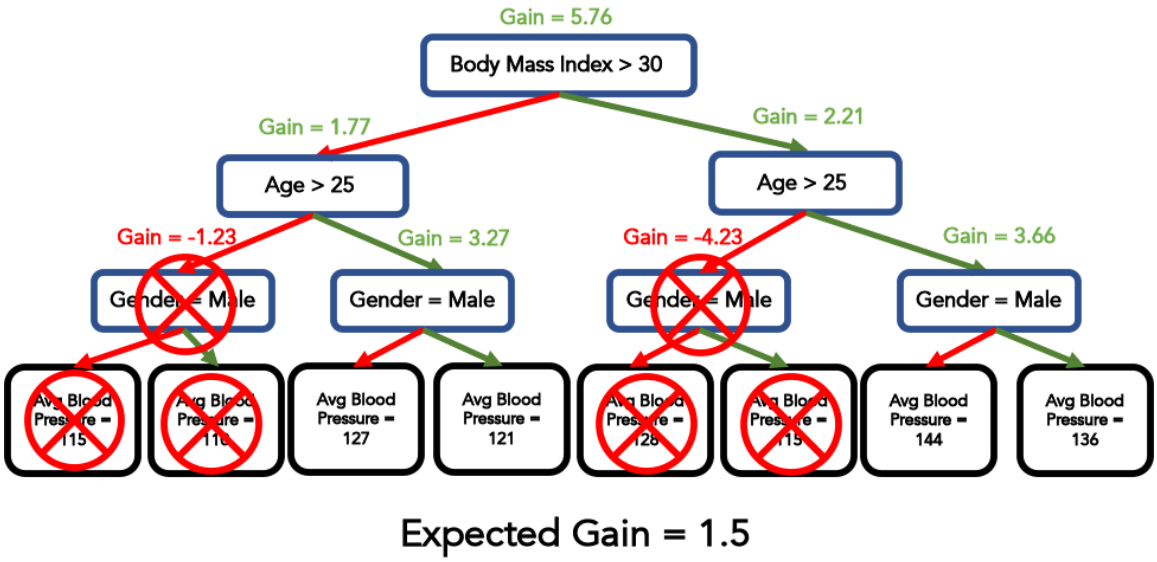
Boosting is a specific type of ensemble technique that focuses on converting a set of **weak learners** into a strong learner. A weak learner is a model that performs slightly better than random guessing. Boosting works by training these weak learners sequentially, where each new model focuses on correcting the **errors or residuals** made by the previous ones. This iterative process continues until a strong model is built.



3. Tree Pruning

Pruning is a very important technique in tree models that helps to prevent overfitting. Eliminate branches in the trees that have little importance or contribute minimally to the model's prediction accuracy. This reduces the complexity of the model by removing unnecessary splits in the trees.

XGBoost uses post-pruning, also known as cost complexity pruning. Initially, the decision tree is grown to its full size to capture all patterns in the training data. Afterward, the tree is pruned by iteratively removing the least important branches or nodes, which are identified based on their minimal contribution to the model's accuracy. This process continues until further pruning no longer improves performance.



For the above figure, if Expected Gain is set to be 1.5 (a hyperparameter), then any tree split with an information gain that is smaller than 1.5 will be removed.

Hyperparameter Tuning

Hyperparameter tuning of XGBoost was conducted using Optuna, a powerful framework designed for multiple machine learning models. Below are the hyperparameters we tuned for this model and their ranges. More information on hyperparameters can be found in the official document of XGBoost.

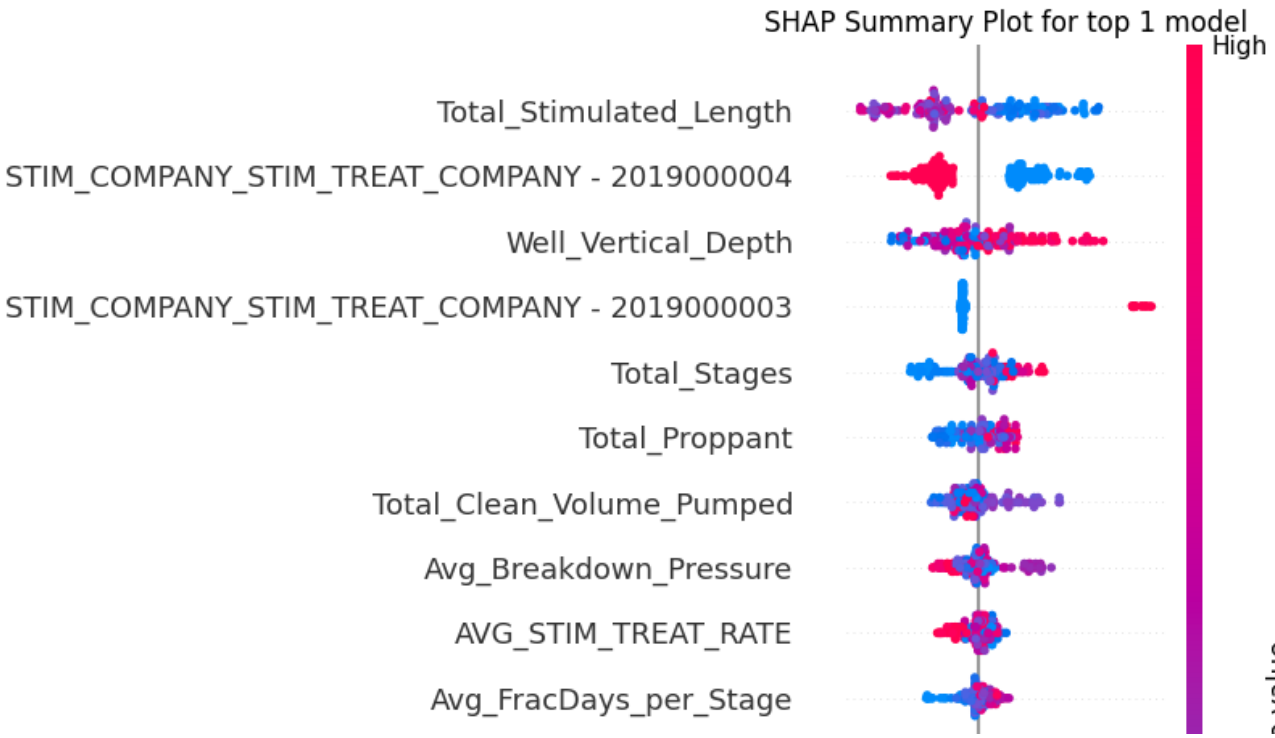
Hyperparameter Name	Range	Definition
objective	req:squarederror	Loss function: squared error
max_depth	[1, 3]	Maximum depth of a tree (weak learner)
learning_rate	[0.001, 0.1] ; log=True	Step size shrinkage used in update to prevent overfitting
subsample	[0.6, 1.0]	Subsample ratio of the training instances.
colsample_bytree	[0.6, 1.0]	The fraction of columns to be subsampled
lambda	[1e-8, 10.0] ; log=True	L2 regularization term on weights.
alpha	[1e-8, 10.0] ; log=True	L1 regularization term on weights.
gamma	[1e-8, 10.0] ; log=True	Minimum loss reduction required to make a further partition on a leaf node of the tree.
min_child_weight	[1, 20]	Minimum sum of instance weight (hessian) needed in a child.

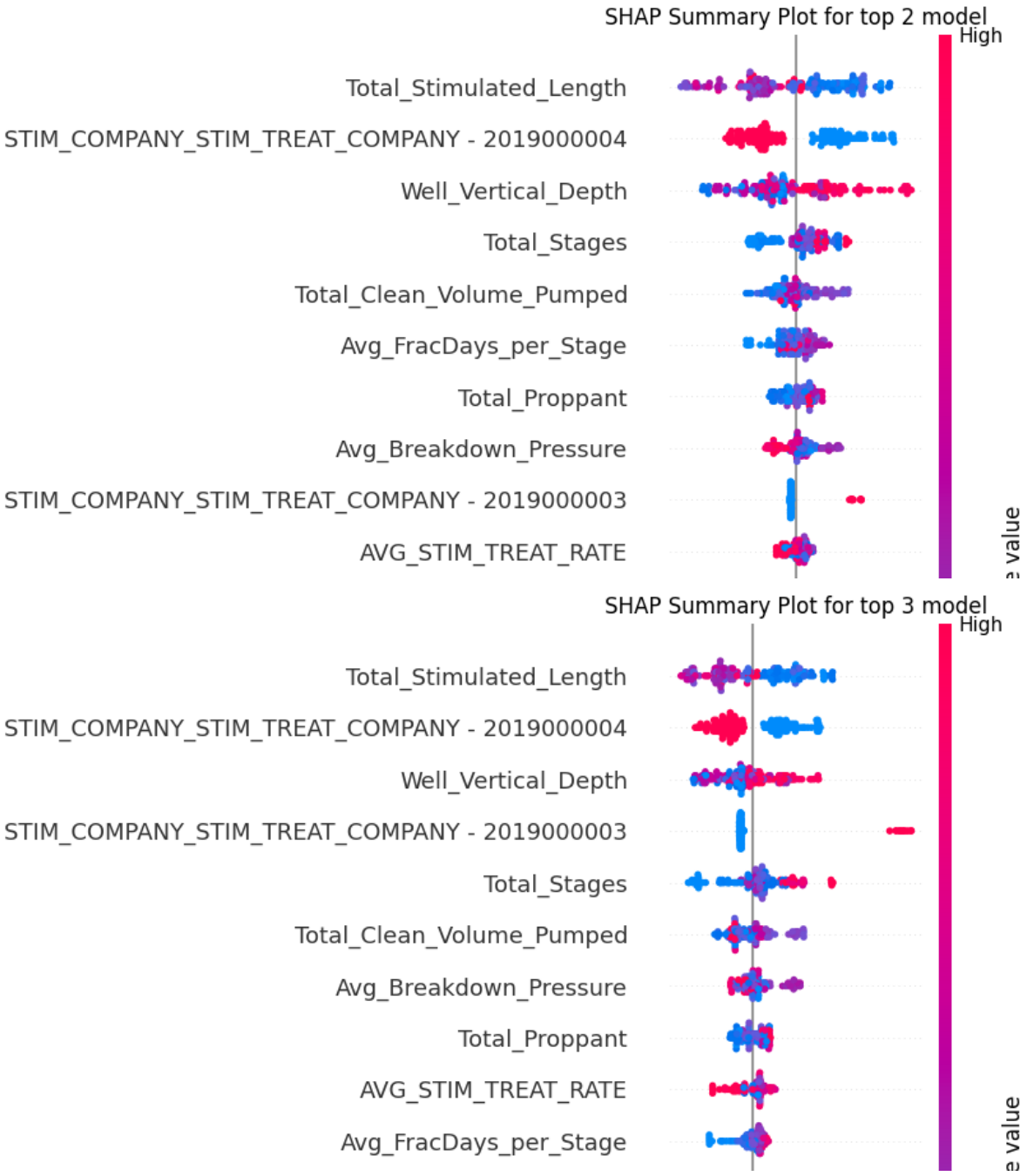
Model Result

Best Hyperparameter (Round to 3 decimal places)
N_estimators = 36
Max_depth = 3
Learning_rate = 0.0837
Subsample = 1.000
Colsample_bytree = 0.887
Lambda = 9.006e-05
Alpha = 0.008
Gamma = 9.743
Min_child_weight = 16

Model Performance
Test set: Median Mean Absolute Percentage Error: 35%

SHapley Additive exPlanations (SHAP)
Top 5 model

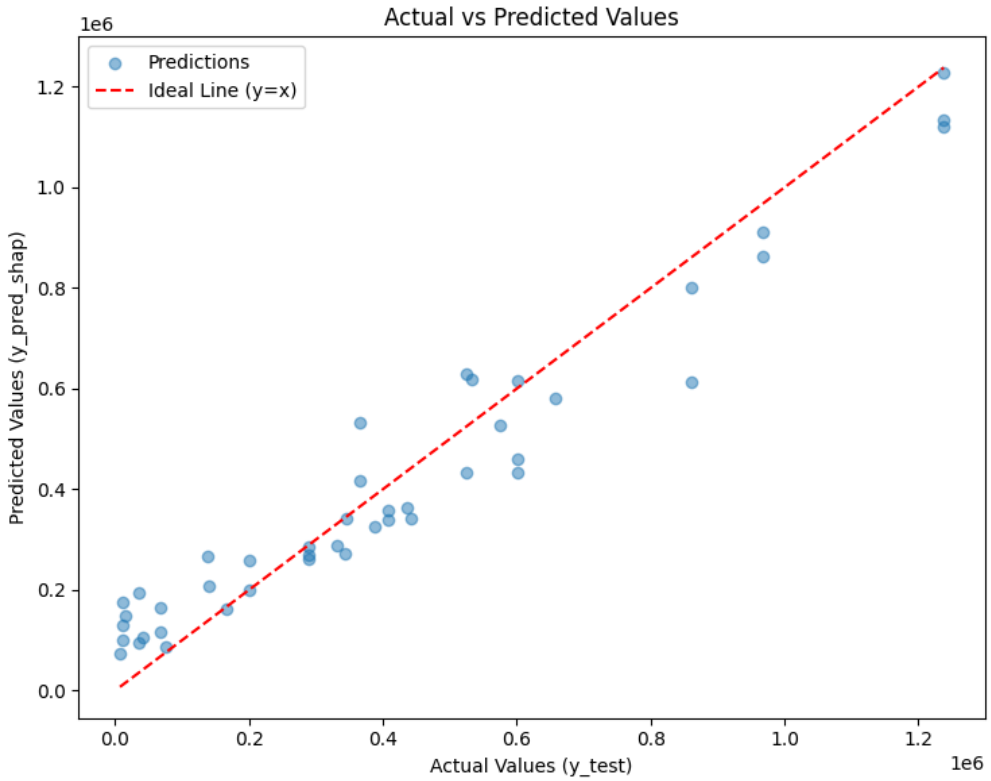



SHAP Summary Plot for top 2 model
Total_Stimulated_Length
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000004
Well_Vertical_Depth
Total_Stages
Total_Clean_Volume_Pumped
Avg_FracDays_per_Stage
Total_Proppant
Avg_Breakdown_Pressure
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000003
AVG_STIM_TREAT_RATE
High
SHAP Summary Plot for top 3 model
Total_Stimulated_Length
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000004
Well_Vertical_Depth
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000003
Total_Stages
Total_Clean_Volume_Pumped
Avg_Breakdown_Pressure
Total_Proppant
AVG_STIM_TREAT_RATE
Avg_FracDays_per_Stage
High

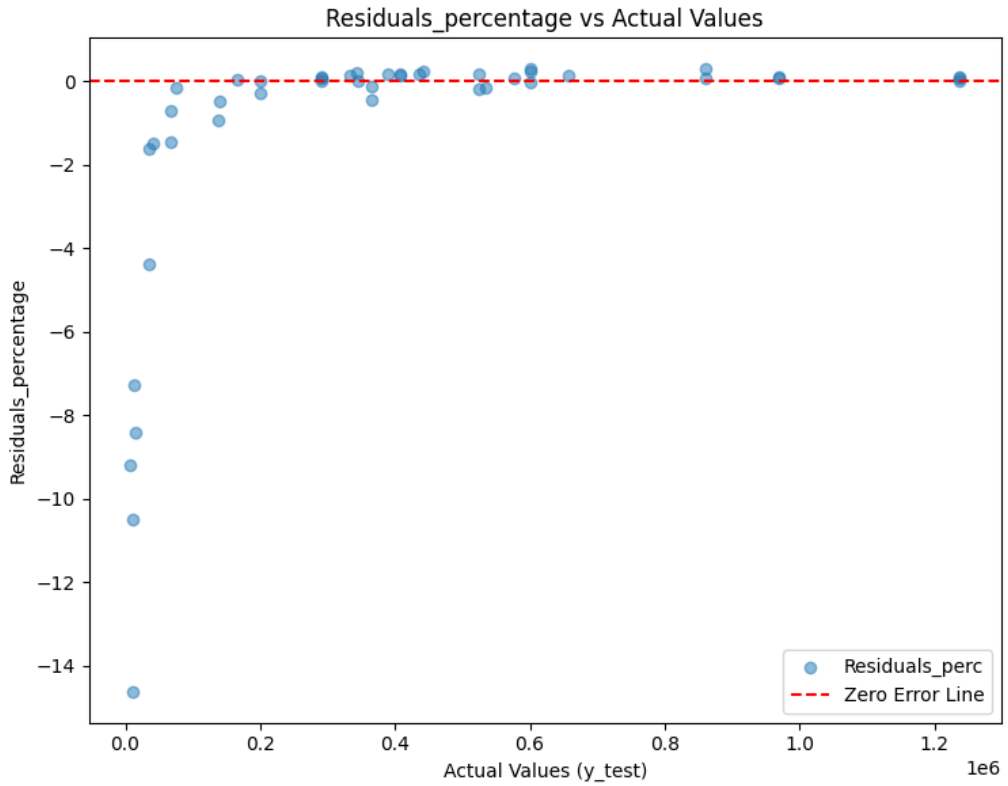

SHAP Summary Plot for top 4 model
High
Total_Stimulated_Length
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000004
Well_Vertical_Depth
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000003
Total_Stages
Total_Clean_Volume_Pumped
Total_Proppant
Avg_Breakdown_Pressure
Avg_FracDays_per_Stage
AVG_STIM_TREAT_RATE
SHAP Summary Plot for top 5 model
High
Total_Stimulated_Length
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000004
Well_Vertical_Depth
STIM_COMPANY_STIM_TREAT_COMPANY - 2019000003
Total_Stages
Total_Clean_Volume_Pumped
Avg_Breakdown_Pressure
Total_Proppant
AVG_STIM_TREAT_RATE
Avg_FracDays_per_Stage

Feature Name	Number of Appearance in Top 10 in SHAP
Total_Stimulated_Length	5
STIM_COMPANY_STIM_TREAT_COMPANY - 201900004	5
Well_Vertical_Depth	5
STIM_COMPANY_STIM_TREAT_COMPANY - 201900003	5
Total_Stages	5
Total_Proppant	5
Total_Clean_Volume_Pumped	5
Avg_Breakdown_Pressure	5
AVG_STIM_TREAT_RATE	5
Avg_FracDays_per_Stage	5

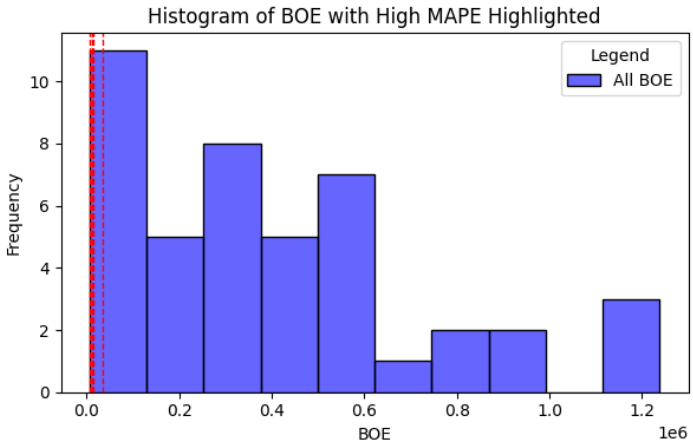
Prediction Analysis

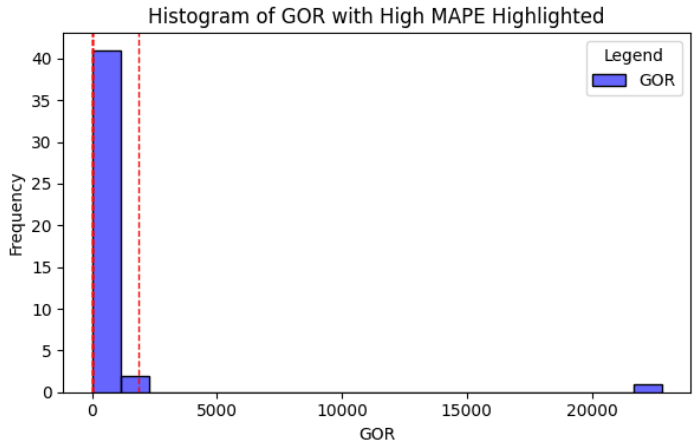


Predicted values are plotted against actual values to compare with the x=y trendline that represents perfect prediction. Based on the plot, data points are scattered around the trendline, showing high prediction performance.



Residuals percentage is plotted against actual values. For low actual value wells (low BOE), the residual percentage was exceptionally higher than wells with larger BOE. The reason is low production level wells have similar completion parameters with high production level wells, but they are the minority of the dataset, making the model incapable of predicting accurately for those wells.





Same conclusion can be seen in the BOE histogram, where the highest 10% MAPE wells are highlighted in red dotted lines. They lie in the low BOE region of the distribution. For gas oil ratio (GOR) histogram, we didn't find any trend for high MAPE wells.
There are two types of reasons why a well produces low BOE:
1. Gas well. (After divided by 6000, the scale is shrinked)
2. Wells with a slow start, but then increased after 6 months or 1 year.

Code References

1. XGBoost_IP180_Final_Submission.ipynb
2.XGBoost_IP360_Final_Submission.ipynb
3. Random Forest.ipynb

Model Selection and Comparison

Metrics Used for Comparison

We evaluated these three metrics to assess model performance comprehensively:

- **Mean Square Error (MSE):** Sensitive to outliers, providing a measure of the average squared difference between actual and predicted values.
- **R-squared (Coefficient of Determination):** Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. However, it can be misleading for non-linear relationships.
- **Median Absolute Percentage Error (MAPE):** Less sensitive to outliers compared to MSE and suitable for comparing prediction errors across different scales for both IP180 and IP360.

Ultimately, we selected **MAPE** as the primary metric for comparison due to its robustness against outliers and its ability to provide a consistent measure of prediction accuracy across different scales.

Model Strengths and Drawbacks

1. Random Forest Model

Pros:

- High Accuracy: Achieves high accuracy due to the ensemble of multiple decision trees.
- Robustness: Less likely to overfit, providing stable predictions.
- Versatility: Capable of handling both regression and classification tasks.

Cons:

- Bias: May struggle with high-dimensional data without proper tuning.
- Computational Cost: Can be expensive and time-consuming, especially with a large number of trees.

2. XGBoost (Extreme Gradient Boosting)

Pros:

- High Performance: Outperforms other models due to the boosting technique.
- Efficient: Utilizes parallel processing, enhancing computational efficiency.
- Good Generalization: Incorporates regularization and pruning to prevent overfitting.

Cons:

- Interpretability: Like other ensemble models, it can be difficult to interpret.
- Sensitivity to Noisy Data: Requires careful data preprocessing to handle noisy data effectively.

Best Model Selection

Selection Considerations

The selection of the best model involves evaluating performance benchmarks and specific criteria including accuracy, and robustness against overfitting.

Random Forest - Model Evaluation

Median Absolute Percentage Error (MAPE) on Test Set:

- **IP180:** ~39%
- **IP360:** ~26%

XGBoost - Model Evaluation

Median Absolute Percentage Error (MAPE) on Test Set:

- **IP180:** ~35%
- **IP360:** ~24%

The XGBoost model shows better performance than the Random Forest model, with lower MAPE for both IP180 and IP360.

Best Model Selection:

The best model selected is **XGBoost** due to its superior performance metrics, particularly its lower MAPE, indicating more accurate and reliable predictions. Additionally, XGBoost's efficient parallel processing and robust handling of data through regularization and pruning contribute to its overall effectiveness and suitability for this project.

Rationale:

- **Accuracy:** Lower MAPE values for both IP180 and IP360 indicate better prediction accuracy.
- **Robust Generalization:** The regularization and pruning techniques in XGBoost prevent overfitting, ensuring that the model generalizes well to new data.

Overall, we observed that the XGBoost model's strengths in performance, and robustness make it the ideal choice for optimizing well completion strategies and predicting production levels in hydraulic fracturing operations.

Applications

Influential Parameters & Definition

Feature Name	Number of Appearance in Top 10 in SHAP
Total_Stimulated_Length	5
STIM_COMPANY_STIM_TREAT_COMPANY - 201900004	5
Well_Vertical_Depth	5
STIM_COMPANY_STIM_TREAT_COMPANY - 201900003	5
Total_Stages	5
Total_Proppant	5
Total_Clean_Volume_Pumped	5
Avg_Breakdown_Pressure	5

AVG_STIM_TREAT_RATE	5
Avg_FracDays_per_Stage	5

We aggregated the SHAP result for top 5 XGBoost models and counted the number of times the most influential 10 features. All 10 features were present in the top 5 models (but with different order, detail please refer to the SHAP plots in XGBoost section), which shows stability in model performance.

The definitions for those top 10 influential features are:
- **Total Stimulated Length:** Total length of all stimulated stages of a well (will be smaller than Well_Lateral_Length), measured in feet
- **STIM_COMPANY_STIM_TREAT_COMPANY - 201900004:** Whether the stimulation process is carried out by companyID "2019000004"
- **Well_Vertical_Depth:** Total vertical length of a well (aka. true vertical depth), measured in feet
- **STIM_COMPANY_STIM_TREAT_COMPANY - 201900003:** Whether the stimulation process is carried out by companyID "2019000003"
- **Total_Stages:** number of stages for that Well (take total count, not the max. of stage_num from data)
- **Total_Proppant:** Total proppant of all stages of a well (including stage1)
- **Total_Clean_Volume_Pumped**: Total clean volume pumped for a well
- **Avg_Breakdown_Pressure**: Average breakdown pressure of all stages of a well
- **AVG_STIM_TREAT_RATE**: Average rate of how fast the hydraulic fracturing fluid is injected into the reservoir formation layer
- **Avg_FracDays_per_Stage**: Average number of days to frac a stage for that well

What-If Analysis

Methodology

This technique helps determine how varying input variables impact a particular output variable under a set of assumptions. In this analysis, we aim to assist engineers in understanding how changes in specific completion parameters affect production volume. This approach is crucial for optimizing well completion strategies by simulating different parameter settings.

To guide this analysis, we have selected three parameters for the what-if scenarios based on the SHAP important features and advice from domain experts.

1. **Total Stimulated Length**:
    - This is the sum of the lengths of all individual fracture stages along the wellbore, excluding the spaces between the stages, as these spaces do not contribute to production.

2. **Total Clean Volume Pumped**:
    - This represents the total volume of fluid (typically water or a clean fracturing fluid) pumped into the well during the fracturing process.
3. **Total Proppant**:
    - This is the cumulative amount of proppant material (such as sand) used during the fracturing process.

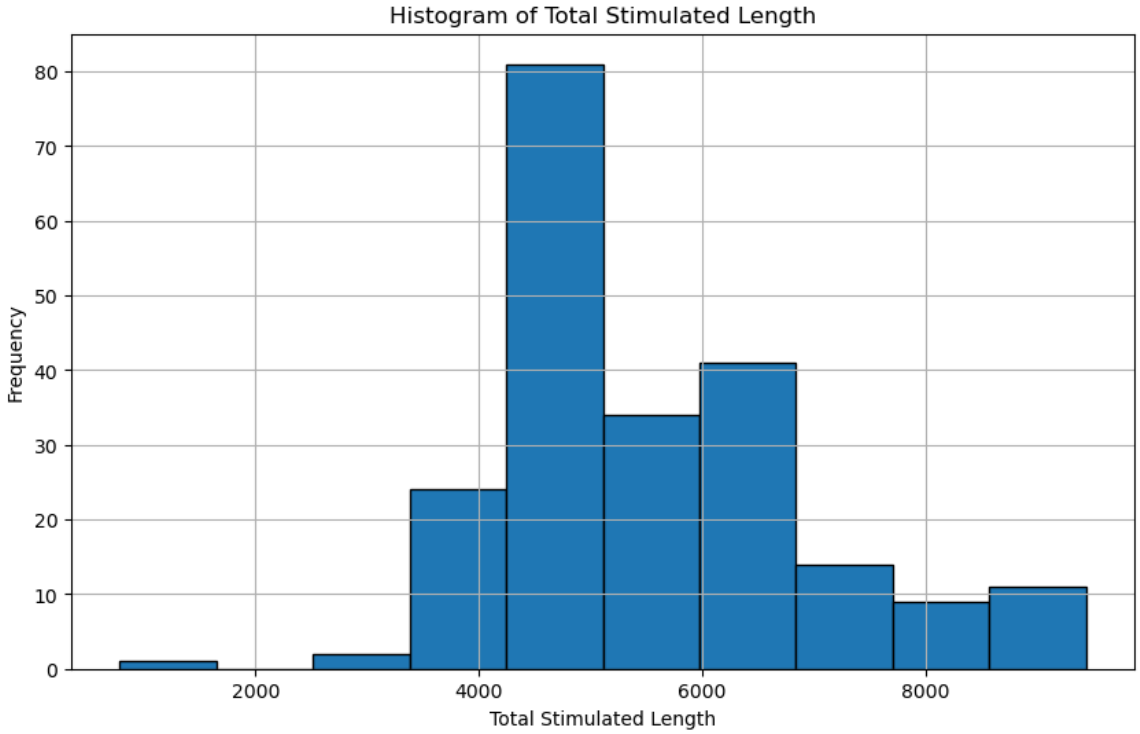
By systematically varying these parameters, we can evaluate their influence on production volume and provide insights for enhancing well performance.

Design

The range for the what-if analysis is determined based on the distribution of data. Due to data coverage challenges, we have to restrict our analysis to these specific ranges
For this design example, we have picked the original values of one median well to illustrate the changes.

What-If design for Stimulated Length

The range of total stimulated length considered is based on the distribution of data, ensuring we capture a realistic variation in production scenarios for this dataset.



Range Considered: [4000, 6000, 8000]

When changing the values of the stimulated length, it is crucial to also adjust the other dependent variables associated with it. In our analysis, these dependent variables include the total proppant used and the total clean volume pumped.

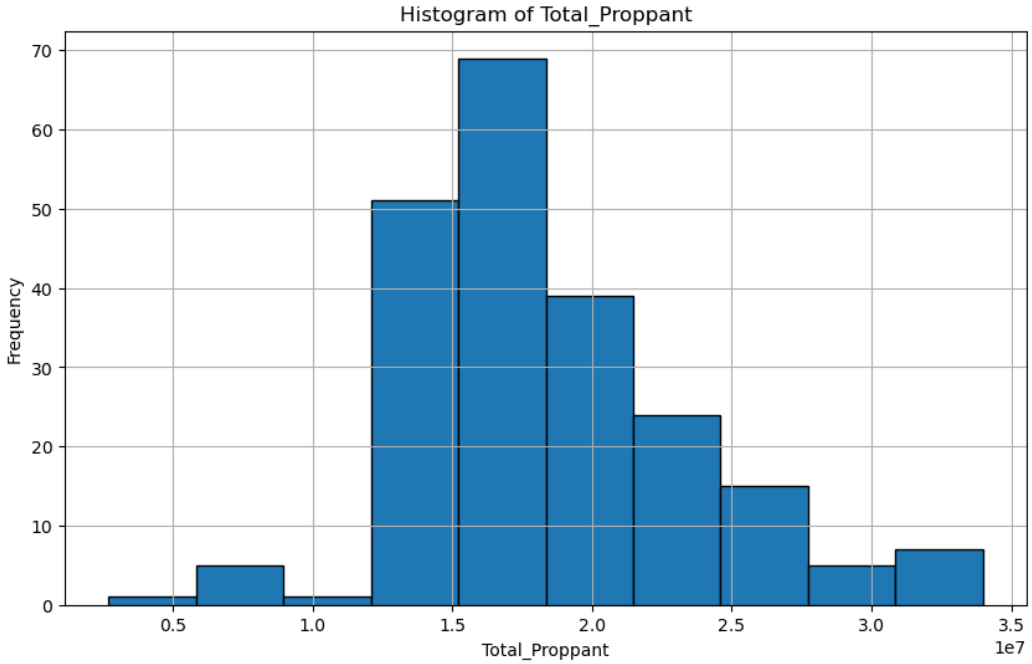
1. **Total Stages**:

- Calculation: Total Stimulated Length / Average Stage Length
- For example, if the average stage length is 144:
  - For a total stimulated length of 4000: 4000/144 ≈ 27
  - For a total stimulated length of 6000: 6000/144 ≈ 42
  - For a total stimulated length of 8000: 8000/144 ≈ 55

2. **Total Proppant**:
   - Calculation: Proppant per Stage × Total Stages
   - The Proppant per Stage for the median well considered in this illustration is 510,953:
     - For a total stimulated length of 4000: 510953 x 27 = 13,795,731
     - For a total stimulated length of 6000: 510953 x 42 = 21,460,026
     - For a total stimulated length of 8000: 510953 x 55 = 28,102,425
3. **Total Clean Volume Pumped**:
   - Calculation: Clean Volume per Stage x Total Stages
   - The clean volume per stage for the median well considered in this illustration is 10057:
     - For a total stimulated length of 4000: 10057 x 27 = 271,539
     - For a total stimulated length of 6000: 10057 x 42 = 422,394
     - For a total stimulated length of 8000: 10057 x 55 = 553,135

What-If design for Total Proppant

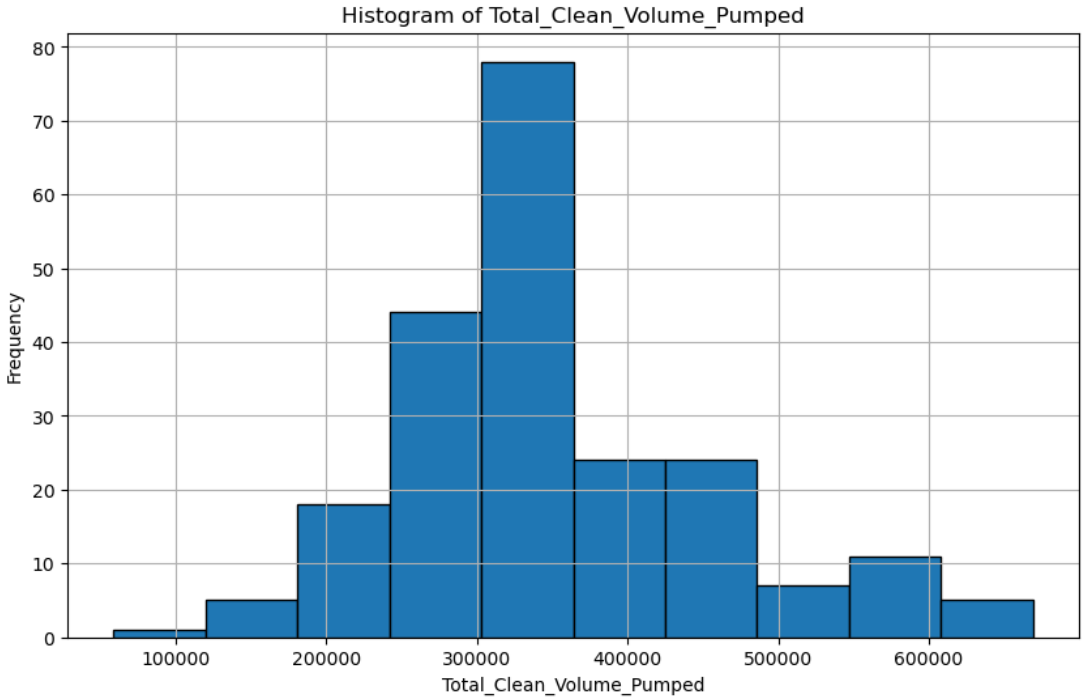
The range of total proppant considered is based on the distribution of data, ensuring we capture a realistic variation in production scenarios for this dataset.



Range Considered: [16,000,000, 20,000,000, 24,000,000]

What-If design for Total Clean Volume Pumped

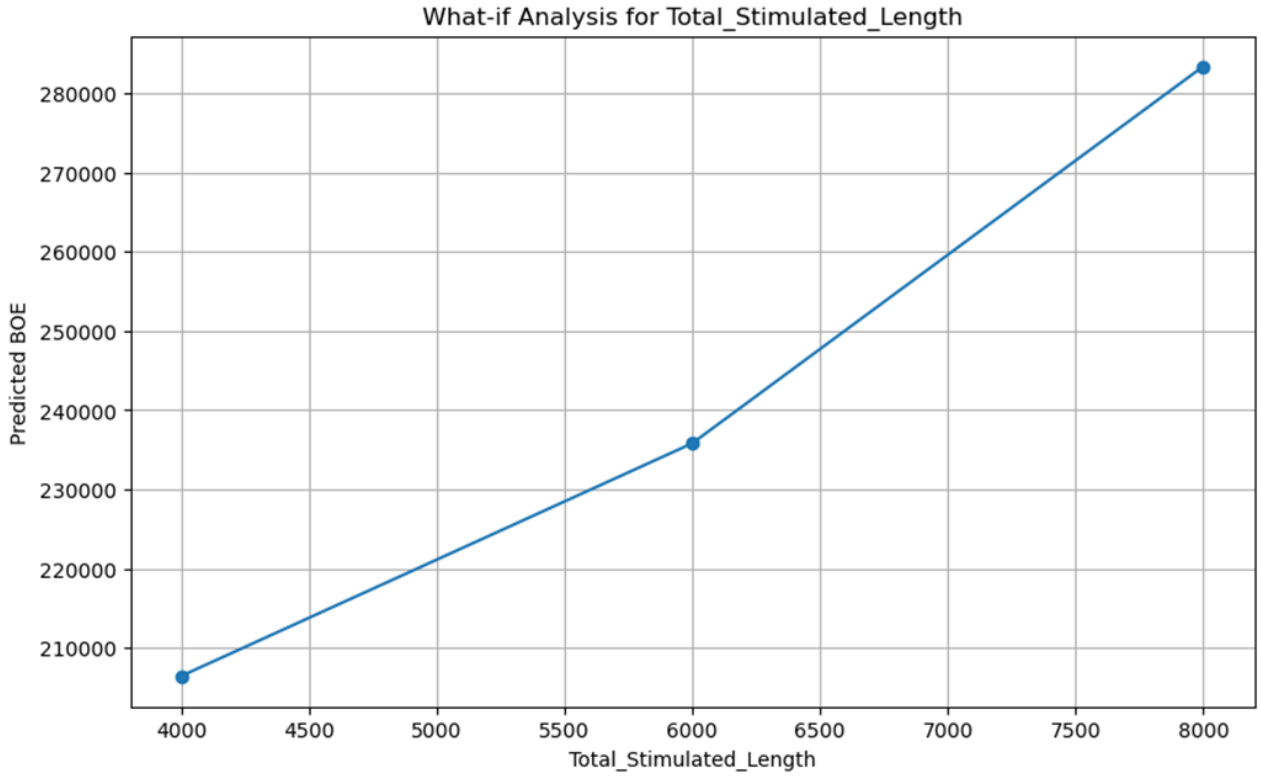
The range of total clean volume pumped considered is based on the distribution of data, ensuring we capture a realistic variation in production scenarios for this dataset.



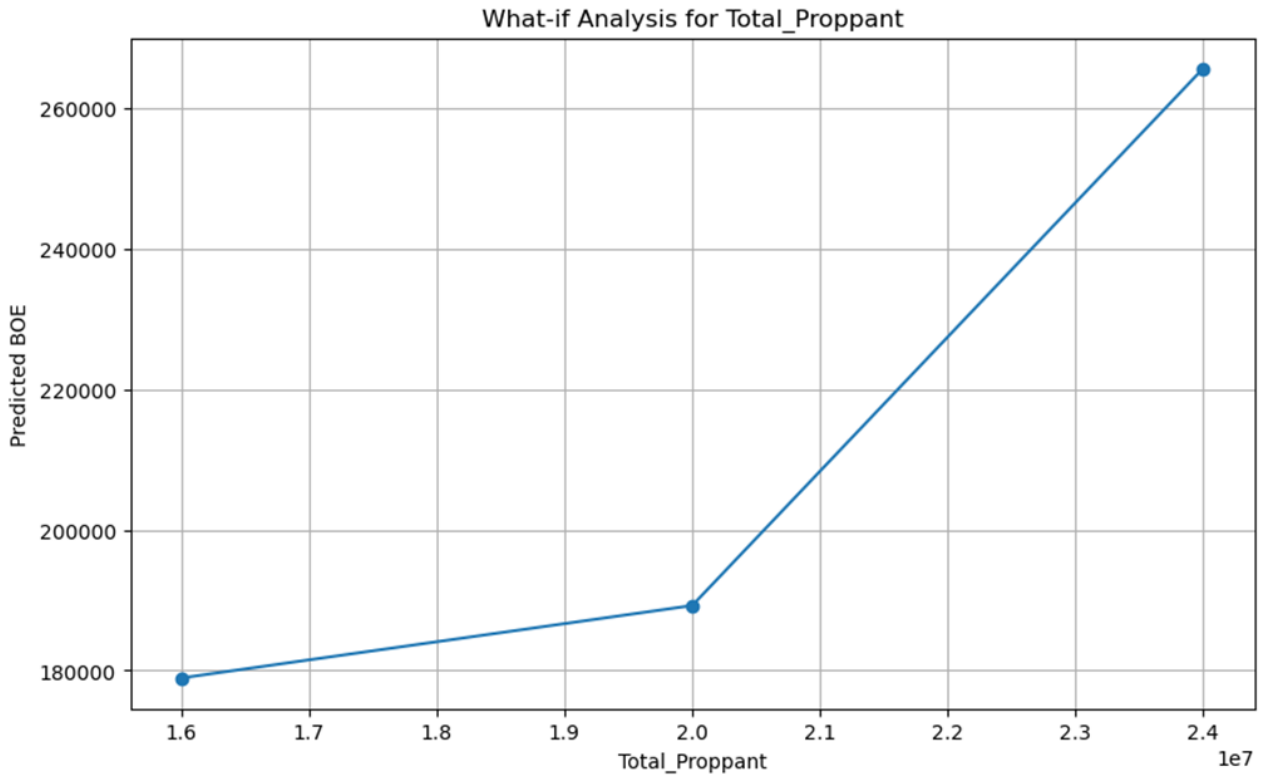
Range Considered: [260,000 , 300,000 , 340,000]

Analysis

Based on the defined ranges for the variables, we performed a what-if analysis on five representative wells.



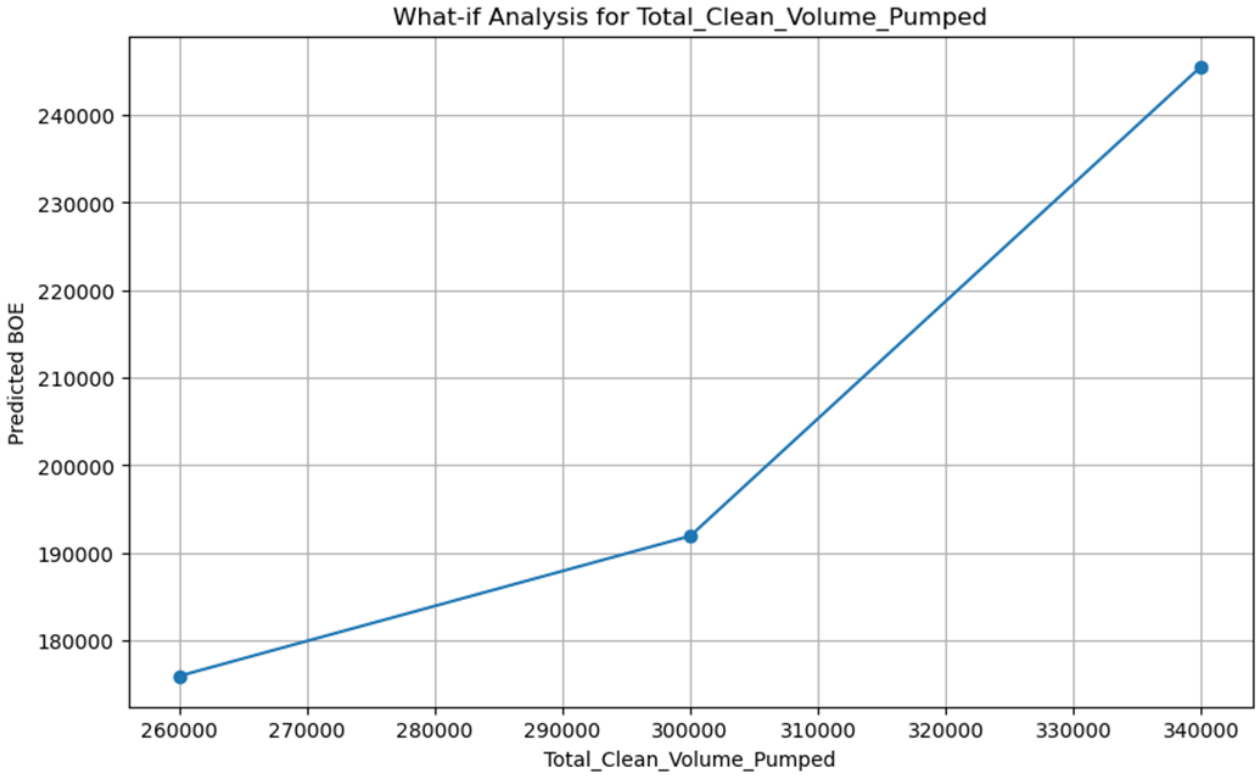
The chart above illustrates the relationship between the total stimulated length and the predicted BOE (Barrels of Oil Equivalent) production.
The X-axis represents the range of total stimulated lengths considered in the analysis, with values plotted at 4000, 6000, and 8000 units. The Y-axis shows the predicted BOE production corresponding to each total stimulated length.
The chart displays three data points connected by a trend line, indicating a positive linear relationship between the total stimulated length and the predicted BOE production.
As the total stimulated length increases from 4000 to 8000 units, the predicted BOE production also increases from approximately 210,000 to 280,000.



The above chart shows the relationship between the total proppant used and the predicted BOE (Barrels of Oil Equivalent) production.

The X-axis represents the range of total proppant used in the analysis, with values ranging from 16,000,000 to 24,000,000 units. The Y-axis shows the predicted BOE production corresponding to each total proppant value.

The chart displays three data points connected by a trend line. The trend line shows a positive relationship between the total proppant used and the predicted BOE production. As the total proppant increases from 16,000,000 to 24,000,000 units, the predicted BOE production rises from approximately 180,000 to 260,000.



The above chart illustrates the relationship between the total clean volume pumped and the predicted BOE (Barrels of Oil Equivalent) production.
The X-axis represents the range of total clean volume pumped, with values ranging from 260,000 to 340,000 units. The Y-axis shows the predicted BOE production corresponding to each value of the total clean volume pumped.
The chart displays three data points connected by a trend line, indicating a positive relationship between the total clean volume pumped and the predicted BOE production. As the total clean volume pumped increases from 260,000 to 340,000 units, the predicted BOE production rises from approximately 180,000 to 240,000.

By systematically varying these parameters within the specified ranges, we aimed to understand their impact on production volumes and identify optimal settings for well completion strategies.

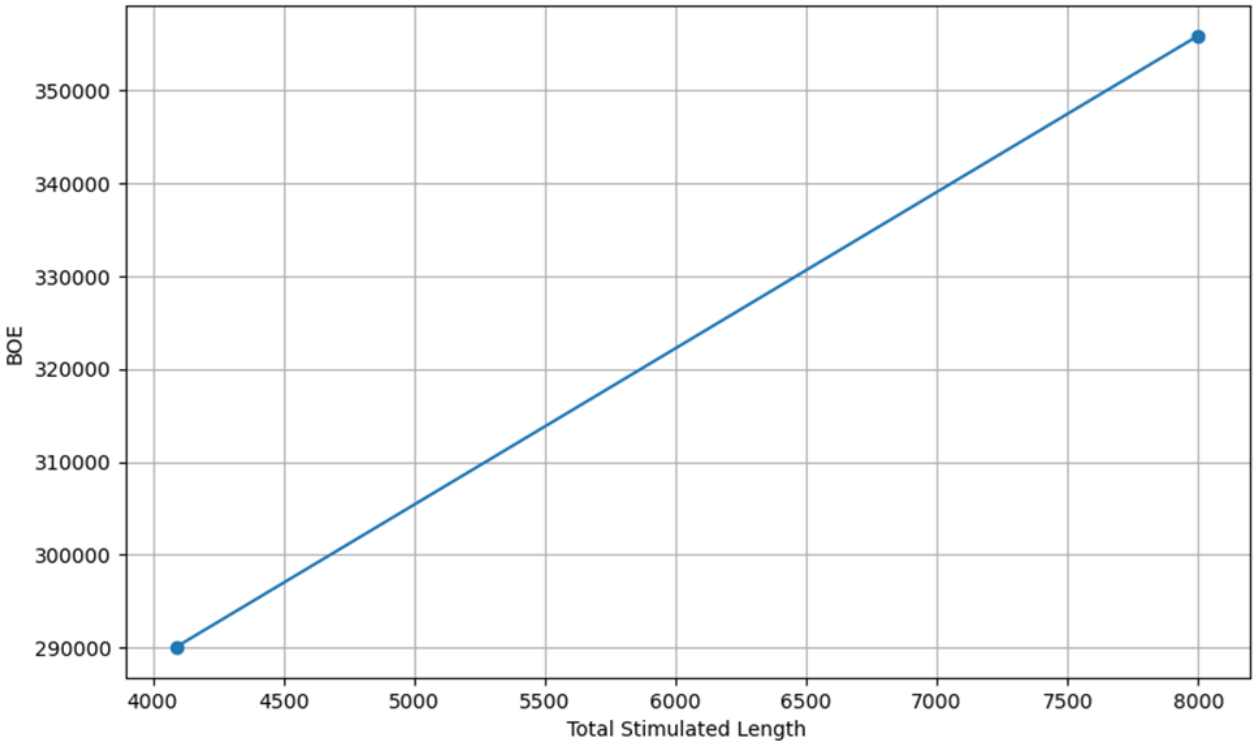
Optimal Completion Strategy

The table below outlines the range of values considered for each variable:

Variable	**Min**	**Max**
Total Stimulated Length	4000	8000
Total Proppant	16,000,000	24,000,000
Total Clean Volume Pumped	260,000	340,000

What we are suggesting here is that if we increase these variables through this range, we would expect to see an increase in BOE. This can be used as an optimization strategy.

For instance, we analyzed a median well with an original total stimulated length of 4090 units and an actual production of 290,091 BOE. Based on the suggested range for total stimulated length, we performed a what-if analysis for stimulated length: what if we had increased the stimulated length to 8000 units, changing its dependent variables as well. The analysis showed that this change would have increased the production to 355,828 BOE, representing a 22% increase.



Based on the results, we have found that our findings are in line with the physical expectations of the model. This alignment reinforces the validity of our approach and suggests practical applications for improving well performance.

It is important to note that these results are specific to the data and model response used in this analysis and cannot be generalized to all data. Each well and production scenario may have unique characteristics that could influence the outcomes differently.

Code Reference

Sensitivity_analysis.ipynb

Future Integration Suggestion

Code Refactoring and Packaging: Refactor the code from the Jupyter notebooks into modular Python scripts. Make a clean separation of data preprocessing, model training, and inference logic. Organize the code into a well-structured Python package, complete with necessary files like `__init__.py` to enable proper module imports.

Environment Setup: Create a `requirements.txt` or `environment.yml` file to list all dependencies required for the project. Write a Dockerfile to containerize the application, ensuring a consistent environment by installing dependencies and setting up the necessary configurations within the container.

Model Serving: Select a model serving framework such as Flask, FastAPI, or Django to create an API, or use specialized tools like TensorFlow Serving or TorchServe. Develop RESTful API endpoints for model inference, including health checks, model predictions, and potentially model retraining capabilities.

CI/CD Pipeline: Set up a CI/CD pipeline using tools like GitHub Actions, Jenkins, or GitLab CI/CD to automate the testing, building, and deployment processes. Ensure automated tests run on each commit, and automate the deployment of the Docker container to a staging environment for further testing.

Deployment: Choose a cloud provider such as AWS, Google Cloud Platform, or Azure to host the application. Deploy the Docker container using Kubernetes, AWS ECS, Google Cloud Run, or a similar service. Implement monitoring and logging using tools like Prometheus, Grafana, ELK Stack, or cloud-specific solutions to maintain oversight of the application's performance and issues.

Conclusions

Successful Development and Implementation of Predictive Models:

- Three machine learning models (Linear Regression with Log Transformation, Random Forest, and XGBoost) were developed to predict the final production amount of oil and gas, measured in Barrels of Oil Equivalent (BOE).
- The XGBoost model was identified as the best-performing model based on its lower Median Absolute Percentage Error (MAPE) compared to other models, indicating higher accuracy and reliability in predictions.

Enhanced Understanding of Influential Parameters:

- SHAP (SHapley Additive exPlanations) analysis provided insights into the most influential parameters affecting production. The top features included Total Stimulated Length, Total Proppant, and Total Clean Volume Pumped.
- These insights helped in identifying the key factors that significantly impact production, aiding in better decision-making and optimization of well completion strategies.

Optimization and Sensitivity Analysis:

- Sensitivity and what-if analyses were conducted to understand the impact of various well completion parameters on production outcomes. These analyses enabled the simulation of different scenarios and provided actionable insights for optimizing operational settings.
- The ability to perform what-if scenarios allowed for the exploration of potential outcomes based on different parameter settings, helping in planning and adjusting strategies to maximize production.

Significant Business Impact:

- The optimized well completion parameters derived from the predictive models and sensitivity analyses are expected to enhance oil and gas production, leading to increased extraction rates and higher output from each well.
- The project demonstrated potential cost savings for ConocoPhillips, estimated to be between $10 to $20 million, by reducing inefficiencies and optimizing operational parameters.

Lessons Learned

- **Domain Understanding**

  Entering the oil and gas industry and applying data science techniques required a deep understanding of the domain. This foundational knowledge was crucial for effectively utilizing data science methods, ensuring that applications were relevant, accurate, and impactful.

- **Data Handling and Challenges**

  Real-life data often presents challenges such as combining datasets with mismatched columns or missing significant portions of information. In these cases, prioritizing data salvage through minor imputation over deletion was key. Feature engineering and exploratory data analysis (EDA) became essential, demonstrating the importance of manipulating data to improve machine learning model performance. The majority of the time was spent exploring and refining the data to ensure optimal outcomes.

- **Variable Evaluation**

  Adjusting approaches based on data characteristics was necessary, as industry preferences influenced variable evaluation. For instance, medians were more appropriate than averages in this project due to the presence of extreme outliers.

- **Time Management**

  Efficient time management was critical, especially when balancing coursework and job searches alongside the project. Treating the project like a professional job emphasized the importance of disciplined time management and prioritization skills, crucial for juggling multiple responsibilities successfully.

- **Communication**

  Effective communication played a vital role in staying organized and on top of tasks. Emphasizing the importance of clear communication channels ensured team alignment, task clarity, and regular progress monitoring. It was also essential to clarify definitions with industry experts to avoid discrepancies and ensure mutual understanding. Limited communication resources required efficient use of available time, such as preparing meeting agendas and consolidating questions.

- **Project Management**

  Flexibility and awareness of absolute deadlines were essential, aiming for early completion to address any unforeseen issues without jeopardizing the overall timeline. Prompt responses to team communications, even brief acknowledgments, were crucial for maintaining alignment and improving efficiency.

Limitations of Current Framework

Dataset Size

We filtered columns from both the completion and production datasets and then performed an inner join. This operation reduced our dataset to only 242 data points. The limited dataset size can lead to several issues affecting model performance.

1. **Overfitting**: The model may learn all the unique patterns and noise in the dataset, causing it to perform poorly on new, unseen data (poor generalization).
2. **Under-representation**: The dataset might not capture all trends and patterns relevant to the business problem, resulting in biased predictions for certain wells.
3. **Inability to utilize advanced models**: Cutting-edge models such as neural networks and large language models require a substantial amount of data to perform effectively. Due to the limited dataset, we cannot fully harness their potential.

Missing Features

We lack crucial geological features that significantly influence the well's production level, such as rock formation. The absence of this geological information leads to sub-optimal predictive model performance.

Geological data can provide insights into the permeability and porosity of the rock, the presence of faults or fractures, and the type of reservoir fluids. These factors are critical for accurately predicting production levels, as they directly affect the flow characteristics and overall efficiency of the well. Including geological information can enhance the model's accuracy and reliability by offering a more comprehensive understanding of the underlying conditions affecting production.

Quality Management Control

1. Weekly Check-Points

   Objective: To ensure continuous progress monitoring, early issue detection, and alignment with project goals.

   - Regular Team Meetings: Weekly meetings with the project team to review completed tasks, discuss upcoming milestones, and address any challenges. These meetings help in maintaining transparency and accountability within the team.
   - Client Meetings: Regular interactions with clients to provide updates, gather feedback, and ensure that the project is meeting their expectations and requirements. This iterative process helps in making necessary adjustments in a timely manner.

- Progress Tracking: Utilize project management tools to track milestones and deliverables. This includes setting clear objectives for each week, documenting progress, and assessing any deviations from the plan.
- Issue Resolution: Early identification and resolution of issues through collaborative discussions. This proactive approach prevents minor issues from escalating into significant problems.

2. Peer Reviews and Testing (Buddy System)

   Objective: To enhance code quality, ensure documentation accuracy, and foster a collaborative learning environment.

   - Peer Review Process: Implement a structured peer review process where team members review each other's code and documentation. This process involves using a standardized checklist to ensure consistency and thoroughness in reviews.
   - Buddy System: Pair team members to review each other's work. This system not only improves the quality of deliverables but also facilitates knowledge sharing and skill development among team members.
   - Early Error Detection: By catching errors early in the development process, the project can avoid costly fixes later on. Peer reviews help in identifying logical errors, code inefficiencies, and areas for improvement.
   - Quality Standards: Establish clear guidelines and standards for reviews to ensure all aspects of the code and documentation are thoroughly examined.

3. Documentation and Code Best Practices

   Objective: To ensure clarity, maintainability, and scalability of the project through well-documented and structured code.

   - Standardized Format: Adhere to a standardized format for documentation and code. This includes consistent naming conventions, commenting styles, and organization of code files.
   - Regular Updates: Continuously update documentation to reflect the current state of the project. This ensures that all team members and stakeholders have access to the latest information.
   - Clarity and Maintainability: Write clear and concise documentation that explains the purpose, functionality, and usage of the code. This practice not only helps current team members but also aids future developers who may work on the project.

- Code Best Practices: Follow best practices in coding such as modular design, avoiding code duplication, and writing reusable functions. This enhances the maintainability and scalability of the project.

Budget

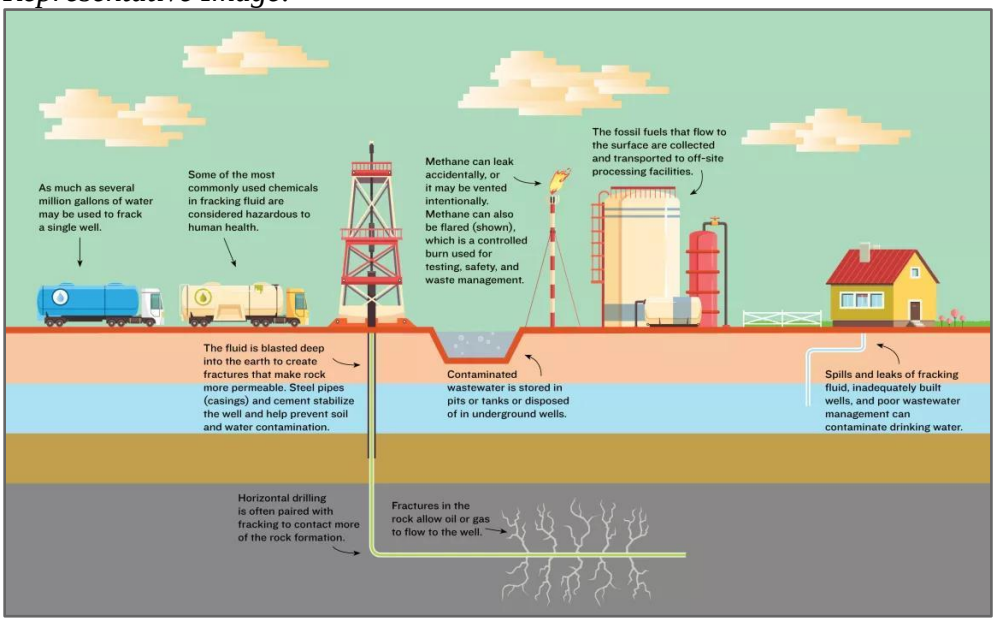
The budget for this project was $45, allocated for an Animaker Starter membership to create the creative video.

APPENDIX A

Glossary

Hydraulic Fracturing Process: This technique is particularly needed to tap into oil reserves in unconventional reservoirs, such as shale rocks,where traditional methods would be insufficient. The process involves high-power fracking trucks injecting a high-pressure viscous fluid mixture(water & some chemicals) into the rock formation to create fractures, allowing hydrocarbons to flow more freely into the wellbore. Trucks having high powered pumps (frac crew) inject pressurized viscous fluid (water and some chemicals) to initiate fractures. This is followed by injection of sand to keep the fracture open and oil flowing.

Representative Image:



Fluid: A mixture of water and chemicals is pumped into the well to create fractures in the rock formation.

Proppant: Mainly sand, it is added to the fluid to keep the fractures open and prevent them from closing under pressure.

Clusters: These are initiation points along the lateral well where fractures are started.

Pump: High-powered pumps are used to inject the fluid and proppant mixture at the necessary pressure to create and maintain fractures.

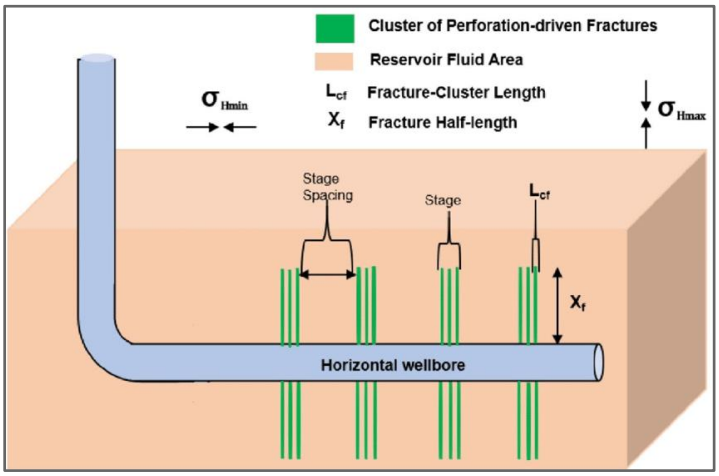
Sidetrack: A sidetrack is a secondary wellbore drilled away from an existing wellbore. This technique is used to bypass an obstruction, access a different part of the reservoir, or re-enter an old well to reach additional reserves.

Breakdown Pressure: The pressure required to initiate a fracture in the rock formation, not a key variable

Average Treatment Pressure: The sustained pressure used during fracturing to keep the fracture open after initiation.

Fluid Pumping: Clean fluid is pumped to break down the formation, while slurry, containing proppant, is pumped to keep the fractures open.

Number of Stages: Refers to the sequential sections along the lateral wellbore that are individually fractured. Each stage is treated separately, allowing for more precise control of the fracturing process and optimizing the distribution of proppant and fluid.



Stage Spacing: Average of Stage Length can be considered as the stage spacing.

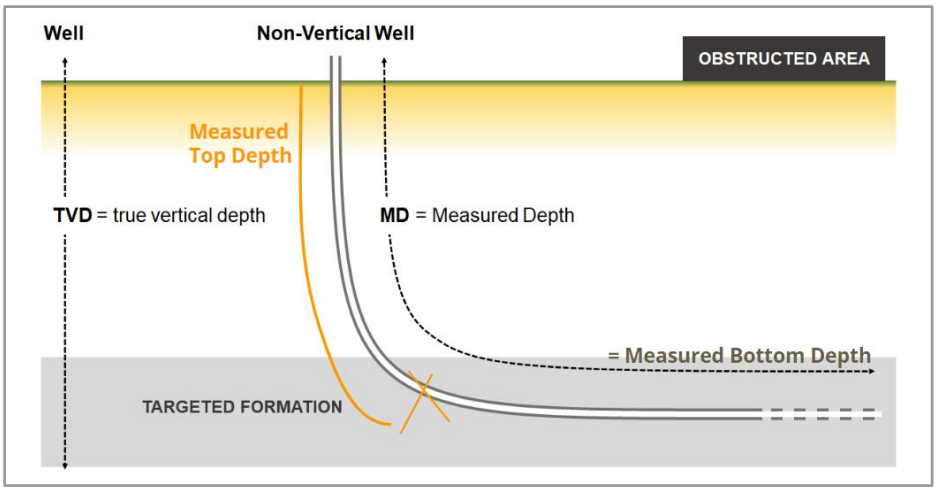
Unconventional Wells: These are wells that use mixtures of sand and shale, which have low permeability and require advanced techniques to enhance production.

Barrel of Oil Equivalent (BOE): Barrel of oil equivalent (BOE) is a way of standardizing natural gas and other energy resources to a barrel of oil's energy. This measurement converts gas production to oil production on an energy-equivalent basis.

True Vertical Depth:
The vertical distance from the wellhead to a point in the wellpath, measured in a straight line perpendicular to the surface. TVD is calculated from directional survey data.

Measured Depth:
The length of the drilled borehole, measured along the path of the wellbore to a feature like a casing point, or geological marker. In perfectly vertical wells, MD equals TVD, but in horizontal or directional wells, MD can be much greater than TVD.

- **Stimulated Bottom Depth:** Measured depth and distance from toe of the well to surface.
- **Stimulated Top Depth:** Measured depth and distance from the first point we hit oil in the well to surface. Always less than bottom depth.



APPENDIX B

Meeting Notes

MEETING INFORMATION:

Objective:	Understanding Project requirements and client expectations
Date:	22nd May 2024, 10 AM - 11 AM ET
Attendees:	ConocoPhillips: Andy Flowers, Amir Nejad, Ben Terrahi Carnegie Mellon: Chris Kowalsky, Gautam Devadiga, Kristen Dmello, Roselle Hsu, Ryan Wang, Vignesh Sridhar
Note taker:	Kristen Dmello

MEETING NOTES:

- **About ConocoPhillips:**
  - Largest independent upstream O&G company that explore for, produce, transport and market hydrocarbons, including crude oil, natural gas, natural gas liquids (NGL), liquefied natural gas (LNG) and bitumen.
  - Traditionally performance is measured on measure of oil in proved reserves. ConocoPhillips has 6.8 Billion barrels oil equivalent proved reserves at year end 2023
- **Background on Unconventional Wells:** These are wells that use mixtures of sand and shale, which have low permeability and require advanced techniques to enhance production.
  More recent advancements to drill unconventional wells:
  - Horizontal Drilling
  - Fracking techniques: Used in unconventional wells with low permeability. Drilling to increase permeability around the horizontal well.
- **Hydraulic Fracturing:**
  - Need: This technique is particularly needed to tap into oil reserves in unconventional reservoirs, such as shale rocks,where traditional methods would be insufficient.
  - The process involves high-power fracking trucks injecting a high-pressure viscous fluid mixture(water & some chemicals) into the rock formation to create fractures, allowing hydrocarbons to flow more freely into the wellbore.
  - Trucks having high powered pumps (frac crew) inject pressurized viscous fluid (water and some chemicals) to initiate fractures.
  - This is followed by injection of sand to keep the fracture open and oil flowing.
- **Problem Statement:** To be able to produce from the newly drilled oil and gas wells, the well should go under a so-called completion process to open up the reservoir to production. Well completion strategy is involved in selecting appropriate combinations of several completion parameters that may be intercorrelated.
- **Current State:** The choices of the completion analysis are driven from the understanding of physics of fluid flow through porous media. The fracture model is focused on Navier stokes' model.
- **Data** collected is from the surface of the wells (in this case one well), this includes features on:
  - Design Elements of fracturing the well
  - Geological Properties of the area
  - Production outcomes at the surface
- **Desired Project Objective:**

Given the data from completion and production from multiple oil and gas producing wells, it is desired to understand the optimal completion strategy to increase the oil production of the wells.
  - The predictive model and/or analytics workflow, should be able to recommend best practices to complete a well.
  - Clarification and justification over the choice of inputs and target variables are required.
  - Since each group of wells are completed with different crew/companies, it is desired to rank the performance of each fracturing company.

- **Evaluation/ Success Criteria:** Ability to answer the below questions:
  - Understand at any given moment which of the 6-8 frac crews is the best performer in North America.
  - What are the key production predictors (IP90, IP180,IP365)
  - Optimal / Best completion design

NEXT STEPS AND ACTION ITEMS:

1. Dataset to be shared by ConocoPhillips (By week of 27th May)
2. Project team to work with client to schedule two placeholders for the week (By 24th May):
   2.1. Office hours with Ben and Amir
   2.2. Weekly touchpoint with Andy for status updates and project milestones
3. Project team to work with client to finalize mid-term presentation date (By 30th June)
4. Project team to research and understand the fracking process and follow up with questions (By 24th May)

MEETING INFORMATION:

Objective:	Q&A with ConocoPhillips to understand the dataset
Date:	30th May 2024, 2 - 3 PM CT
Attendees:	ConocoPhillips: Andy Flowers, Amir Nejad, Ben Terrahi Carnegie Mellon: Chris Kowalsky, Gautam Devadiga, Kristen Dmello, Roselle Hsu, Ryan Wang, Vignesh Sridhar

MEETING NOTES:

- Estimated impact of this project:
  - MVP for up to 8 wells: $100K with optimizing variables
  - Scaled Production for up to 50-60 wells: Few Million dollars savings with optimizing variables
- **Benchmark of model operational efficiency:** ConocoPhillips has evaluated multiple models from regression to MVA analysis in the past. There is no single unique solution to select design parameters. The benchmark can be set as any selected Naive baseline model in this scenario.
- **Data walkthrough:**
  - We need to refer to two datasets provided for the purpose of this project:
    i. Cross_bu_dnc_stim_stage_detail
    ii. Monthly_production
  - Data provided is proprietary data of ConocoPhillips and has been anonymised for this project.
  - Feature types:
    i. Engineering features: Can be decided and changed by the team
    ii. Geographic features: Not defined
  - Use UWI and Highest UWI Sidetrack to get a unique identifier.
  - PRIMARY_JOB_TYPE - Do not use recompletion data, and stick with initial completion data as part of the scope of this project.
  - Production start date indicates when you struck oil. Sometimes there is a lag between job start date and production start date, this can give some insight.
  - Focus on all variables in Drilling & Completion operations for the MVP. Other variables can be considering during the production and refining stage of the project
  - STIM_INT_TREAT_TYPE: Filter out data related to fracturing for the scope of this project. Eg. Acidizing, Acid Matrix
  - STIM_COMPANY, STIM_DIVERSION_COMPANY: Fracking crew data anonymised
  - STIM_INT_DELIVERY_MODE,STIM_INT_EST_SAND_TOP_DEPTH, STIM_INT_DIAGNOSTIC_METHOD, STIM_INT_FRAC_GRADIENT - Ignore these as they're not valuable for the current analysis
  - Post-drill features: These are known after drilling the well
    i. BREAKDOWN_PRESS: The pressure required to initiate a fracture in the rock formation, not a key variable
    ii. STIM_INT_TREAT_AVG_PRESS The sustained pressure used during fracturing to keep the fracture open after initiation.
    iii. STIM_INT_INITIAL_SHUT_IN_PRESS
  - Pre-drill features: These are known before drilling the well. Eg. Geology
  - STIM_INT_PROPPANT_TOTAL is a key design variable measured in pounds per stage
  - STIM_INT_TREAT_AVG_RATE - Pump rate and a key design variable
  - STIM_INT_STAGE_NUMBER: Group by UWI and find how many stages are there in each well. Perform aggregation
  - STIM_INT_TOP_DEPTH and STIM_INT_BTM_DEPTH: Measured depths vertical and horizontal. Might correlate to the lengths of the lateral of the well.
  - STIM_INT_CLEAN_VOLUME_PUMPED and STIM_INT_SLURRY_VOLUME_PUMPED: Clean fluid is pumped to break down the

formation, while slurry, containing proppant, is pumped to keep the fractures open.
    - STIM_INT_BOTTOM_DEPTH_TVD and STIM_INT_TOP_DEPTH_TVD: Can help understand cluster and stage spacing of fracturing.
    - LEASE_OIL_PROD_VOL: Total oil or gas (OIL_GAS_CODE) for each month(CYCLE_MONTH)
    - Delete months where oil production is zero and wells have stopped producing oil. Oil production data can be aggregated at 6 months, 1 year, 2 years, and 5 years.
    - We can choose to build separate models for Oil and Gas production and for each aggregated time period. Alternatively, we can combine the two using the Barrel of Oil Equivalent (BOE) calculation:
        i. Oil wells: Oil & Gas BOE
        ii. Gas wells: Gas & Liquid BOE - Will need to convert gas volume to liquid oil volume

- Process Walkthrough and definitions:
    - Sidetrack: A sidetrack is a secondary wellbore drilled away from an existing wellbore. This technique is used to bypass an obstruction, access a different part of the reservoir, or re-enter an old well to reach additional reserves.
    - Fracturing Process:
        i. Fluid: A mixture of water and chemicals is pumped into the well to create fractures in the rock formation.
        ii. Proppant: Mainly sand, it is added to the fluid to keep the fractures open and prevent them from closing under pressure.
        iii. Clusters: These are initiation points along the lateral well where fractures are started.
        iv. Pump: High-powered pumps are used to inject the fluid and proppant mixture at the necessary pressure to create and maintain fractures.
        v. Number of Stages: Refers to the sequential sections along the lateral wellbore that are individually fractured. Each stage is treated separately, allowing for more precise control of the fracturing process and optimizing the distribution of proppant and fluid.

NEXT STEPS AND ACTION ITEMS:

1. Project team to work with client to schedule two events:
    1.1. Office hours with Ben and Amir (Schedule on calendar once a week)
    1.2. Weekly touchpoint with Andy for status updates and project milestones - To be discussed
2. Project team to work with client to finalize mid-term presentation date (By 7th June)
3. Project team to start researching the fracturing process and working on the Scope Document (By 31st May)

MEETING INFORMATION:

Objective:	Q&A with ConocoPhillips to understand the dataset
Date:	6th June 2024, 2 - 3 PM CT

MEETING NOTES:

- Scope Clarification: Both the economic analysis and the fracturing company ranking would be nice to have, and can be carried out taking into consideration the time availability during the project.
- Dataset Overview:
    - The dataset covers one basin, but there's no specific indicator for the region or basin.
    - After aggregations and cleaning, there are 530 wells in the cross-production dataset and 327 wells in the monthly production dataset.
    - Some wells Some of the data of wells missing in the production data may be because the wells belong to other companies and do not measure production quantities.
    - Handling Missing Values: Instead of removing missing values, adjust them to prevent overall record loss. Make an educated decision about outliers in the features or different null values in the features.
- Aggregating Completion Data:
    - UWI Sidetrack:
        - UWI follows the API 10 convention of identifying these wells and is a code that gives information on the state code, county and latitude/longitude.
        - Check if any UWIs have more than one sidetrack (UWI-Sidetrack).
        - Use the maximum sidetrack for completion data and ignore records associated with lower sidetracks.
    - We need one row per well, for completion data. Right now we have multiple stages(rows) for each UWI. We need to aggregate each design parameter across the stages for each UWI and consolidate them in a single row.
- Aggregating Production Data:
    - Select a target variable (e.g. IP60, IP180, IP360) representing cumulative production after a specific number of days (60,180,360).
    - Discard records where both oil and gas production volumes are zero.
- Feature Engineering:
    - Calculate STIM_INT_BOTTOM_DEPTH (Measured depth and distance from toe of the well to surface).
    - Calculate STIM_INT_TOP_DEPTH (Measured depth and distance from the first point we hit oil in the well to the surface. Always less than bottom depth).
    - Compute Stage_Length (STIM_INT_BOTTOM_DEPTH - STIM_INT_TOP_DEPTH).
    - The SUM of Stage_Length represents the length of the lateral (horizontal section where production occurs) = STIMULATED_LATERAL_LENGTH.
    - Alternatively, the AVERAGE of Stage_Length can be considered as the stage spacing.
    - Calculate NUMBER_OF_STAGES as Count of STIM_INT_STAGE_NUMBER.

NEXT STEPS AND ACTION ITEMS:

1. Project team to share Scope Document for Client's review (By 12th June)
2. Project team to review Data Aggregation performed with Client's team (13thJune)

MEETING INFORMATION:

Objective:	Office Hours - Discuss Modeling Results
Date:	11th July 2024, 3 - 4 PM EST

MEETING NOTES:

- Final presentation will be in person at 9 AM EST, Andy will confirm details
- Use Mean Absolute Percentage Error to evaluate the models, it incorporates scale for both IP180 and IP360
- Utilize MLFlow for Model Logging and to create a simple server for models and metrics performance comparison (Optional)
- XGBoost Feedback:
    - MAPE is a very large number because there's a BIG outlier - Need to treat it
    - Understand which specific well # have the high value in graph of y_pred on y_actual and has created the outlier
    - Understand if the values are for low producing wells or high producing wells - look at the cross plot of (y_test on x) vs (y_pred on y_actual)
    - Geological data missing has created a lower performance, hence we can't meet the 20% MAPE golden standard
- SHAP Interpretation:
    - Well depth: A lot of oil wells are present in our data and hence lower values of depth are creating more impact on the target variable. Deeper wells have higher temperature and produce more natural gas.
    - Fluid viscosity: Adding more propant to a reducing fluid results in a better production - Add a disclaimer: We know that there's a lot of physical constraints involved but we don't have that information and we don't have the data that quantifies that.
    - Lateral length: Specifies how much we're contacting the reservoir - This SHAP result is counter-intuitive since we would expect a linear increase - The higher lateral length should mean you're accessing more of the reservoir but this does not make sense in our result
- Suggested Next Steps:
    - Analyze % error well by well and rank the wells
    - Evaluate if there's an issue with the data for that well that has the higher error %
    - If that well is erroneous in all of the models and cannot be explained, then we need to remove it from the dataset but this would be the last resort. We need to first investigate the data.
    - For all the folds (training, validation and test) in different color schemes - Plot y_pred vs y_actual and out X=X as the red penalizing line - Shows how bad the model is
    - For all the folds (training, validation and test) in different color schemes in residual plot
    - Calculate MAPE on each fold (training, validation and test)
    - Experiment with removing some of the features which are at the bottom of SHAP, this should potentially improve the model
    - Remove Stage 1 information since it's uncorrelated with the rest of the well performance - It's the hardest part of the fracking process and it won't be correlated to the rest of the process

NEXT STEPS AND ACTION ITEMS:

1. Create a presentation outline and get feedback during the next meeting
2. Incorporate modeling changes based on feedback provided by Amir

MEETING INFORMATION:

Objective:	Q&A with ConocoPhillips to discuss modeling
Date:	16th July 2024, 11 - 11:30 AM EST
Attendees:	ConocoPhillips: Andy Flowers, Amir Nejad, Ben Terrahi Carnegie Mellon: Chris Kowalsky, Gautam Devadiga, Kristen Dmello, Ryan Wang, Vignesh Sridhar

MEETING NOTES:

- Overall Feedback: Great approach and this aligns with their current approach. We should propose looking into these wells deeper to understand operational
- Threshold is 100K for IP180 is too large, instead we should try for 10K or 20K is more realistic
- Plot distribution of BOE's on a histogram and highlight where the high error wells are falling using a vertical line to highlight the location of the high error wells.
- Causes for high error wells: Downtime, Capacity of pipeline, choking the well or data problem
- Error metric suggestions and suggested plots:
    - 4 metrics - Median and Mean APE for test and train
    - Calculation of MAPE - Mean and median
    - Actual % error and a function of GOR (Gas over Oil Ratio) - See wells that are having high GOR and check if that explains the error
    - If we drop outliers, we can make the plot a square
    - Find out why different metrics and huge gap between train and test MAPE - Calculate the train MAPE during the hyperparameter tuning
- XGBoost Modelling feedback:
    - Rerun the top 5 models and calculate the test MAPE, if it's consistently that low then the model is overfitting
    - 10 is high in terms of MaxDepth - Maybe 5 or 4
    - Number of trees stopping criteria - Use num of Estimators as 500 and Early Stopping function of XGBoost in training process
      train(..., evals=evals, early_stopping_rounds=10)
      Python Package Introduction — xgboost 2.1.0 documentation
    - From the top model pick the one that has the most balance between train and test MAPE - OPTUNA has the capability to save the auxiliary output for each of these trials (post line 33 create train MAPE - Median and Mean)
      trial.set_user_attr("train_mape", train_mape)
    - Cross validation mean and standard deviation can be evaluated too
- Univariate Sensitivity Analysis and What-If Analaysis:
    - Start doing the sensitivity with the completion parameters on existing models
    - Pick a well or 5 typical wells
    - Look at distribution of prod and see median
    - Make a change in parameter - forecast production and plot vs completion design
    - Eg. If we're changing lateral length - be consistent - also change the total propant too
    - Ben and Amir mentioned they will help us to come up with designs based on the final model feature importance

MEETING INFORMATION:

Objective:	Q&A with ConocoPhillips to discuss modeling updates
Date:	18th July 2024, 3 - 4 PM EST
Attendees:	ConocoPhillips: Amir Nejad, Ben Terrahi Carnegie Mellon: Gautam Devadiga, Kristen Dmello, Ryan Wang

MEETING NOTES:

XGBoost:

- Only CV Median MAPE to be used for the evaluation and sorting. Use the train and CV difference on Median. Don't add the standard deviation because it would be a very low number
- Do the post-processing visually for the top 10 models and look at all the metrics for them. If train MAPE and standard deviation around mean is then low
- Drop 'STDEV_PROPPANT_PER_STAGE_exclu1', 'STDEV_STIM_TREAT_RATE', 'Avg_Space_Between_Stages'
- Make the metrics consistent, either fraction or %
- Show outlier wells visualized with rationale as to why we removed them before the second modeling phase
- Compare SHAP feature importance of top 10 features of top 5 models - Which parameters are changing vs staying the same
- Choose model from the above 5 and do a sensitivity analysis
- GOR calculations - Divide Gas by Oil and multiple by 1000
- Try 5 % and 10% quantile, other than that we're losing information

Sensitivity Analysis:

- Try for features: Propant per stage, Stage Length or Stage Number
- Design a loop and create a function that takes the value of the completion data and makes a prediction
- The histogram of each feature is the guide to try 2-3 different values
- Lateral lengths - 5000 feet is the baseline, 7500,10000 : Compare the production increase
- We need to change the features consistently. Eg. Stage Lengths vs Lateral lengths (Use a plot between two variables and see the linear trend lines). Make bar chart with cases and Y axis with predicted BOE
- Ben's suggestion on baseline and changes:
  standard LL = 5000
  standard stg length = 200
  standard prop/stg = 300k - 500k
  stgs = 5000/200
  LL baseline 5000: 7500 and 10000
  baseline stg length 200: 100, 300

MEETING INFORMATION:

Objective:	Q&A with ConocoPhillips to discuss modeling
Date:	23rd July 2024, 4:30 - 5:30 PM EST
Attendees:	ConocoPhillips: Amir Nejad, Ben Terrahi Carnegie Mellon: Chris Kowalsky, Gautam Devadiga, Kristen Dmello, Ryan Wang, Vignesh Sridhar

MEETING NOTES:

Overall feedback:

- Histogram should be built only on training data, do not use the test data to select baseline value.
- Do some stratify splitting instead of random state when sampling: We can use the sklearn library to stratify over production data in lateral lengths to ensure your training set has a representative of all the quantiles of the lateral lengths.

```
from sklearn.model_selection import StratifiedShuffleSplit
split = StratifiedShuffleSplit(n_splits=10, test_size=0.2, random_state=42)
for train_index, test_index in split.split(df, df['Categories']):
strat_train_set = df.loc[train_index]
strat_test_set = df.loc[test_index]
```

  python - sklearn stratified sampling based on a column - Stack Overflow

- For sensitivity analysis include all the wells, let the model see all the possibilities in the data. Ensure the model does not overfit to any single well. We can pick 2-3 representative wells (20 in reality) and then create a plot which would avoid overfitting to a single well.
- The recommendations can be made based on the trend from the what-if scenarios

What if Scenarios:

- **Average Stage Length:**
  - Trend wise this is not what we expect, it does show sensitivity but the direction of it is not aligned. Stage lengths are tricky
  - Having total stages and stage length is redundant - one of them is enough to capture the stage specification or stage properties. Since the total stages graph is relevant it seems like if we remove stage lengths from the model it should help get a more accurate graph.
  - DROP Stage Length from the model and do the sensitivity analysis
  - Stage Property - Either the stage length or number of stages. Whenever we do perform the sensitivity the lateral length should remain constant (Eg. 7000). While we change total stages we change stage length.
  - For stage related sensitivity do it versus stage length, don't do it based on total stages. It's okay if you don't have stage length in the model. Divide your lateral length by 180 and that becomes the number of stages that we need to feed to the model. Total Stages isn't familiar and we usually keep stage length (29:20)
- **Well Lateral Length:**
  - Don't go as high as 10K - Baseline should be 4K,6K,8K
  - It needs to increase somehow, the longer lateral you have the more access you have to the reservoir
  - 7500 to 10000 is performing as we'd like it to
- **Total Proppant:** 10 - 50 Million baseline (Instead of per stage)
- **Fluid Viscosity:** 1.2 or 1.3 baseline

Next steps:

- First try the well lateral length, proppant and fluid sensitivity . If there are concerns with the above graphs once created then we might have to do the average stage lengths
- **Changes to the dataset:**
  - Convert Average Proppant per stage to Total Proppant
  - Remove the Average Space between stages and average stage length
  - Total Fluid (ie. Total Clean) = Average Fluid (ie. Average Clean) * Number of Stages, also remove the average fluid
  - Total Proppant = Proppant per Stage * Number of Stages
  - Lateral Length to be replaced by Stimulated Lateral Length (context added below in definitions)

- **Changes to the model:**
  - Stratified Sampling
  - Use the best hyperparameter to train the model based on the whole dataset. This is the final model which will be used to

Definitions:

- Lateral Lengths: Top of the lateral - Bottom of the lateral for the horizontal lateral
- Stage Lengths: Top of stage - Bottom of stage
- Lateral Lengths / Stage Lengths = ~Total of Stage Lengths across all stages
- Stimulated Lateral Length = Total of Stage Lengths across all stages
- Unstimulated Lateral Length: Ideally there should be no spaces between the stages but there are sometimes some cases where we leave the area unsimulated. That's why there is a slight different between (Lateral Lengths / Stage Lengths) and Total Stages
- But we should assume and use the Total of the Stage Lengths as the Lateral Lengths

Next Steps:

- Share the document with results once we have updated our approach and Amir and Ben will respond