S&P Global
Commodity Insights

295660 - Data science Technical assessment: Gas production prediction.

Alberto M. Palacio Bastos

April 2024.

Presentation by

Alberto Palacio
M.Sc. Data Scientist
alberto.palaciob@gmail.com

Summary

Data enthusiast with **10+ years of experience** in engineering and data projects for the construction, mining, energy, and oil & gas industries.

Proficient in data analysis and extracting insights applying the **CRISP-DM** (Cross-Industry Standard Process for Data Mining) and **EDA** (Exploratory Data Analysis) methodologies for intelligent data driven decision making.

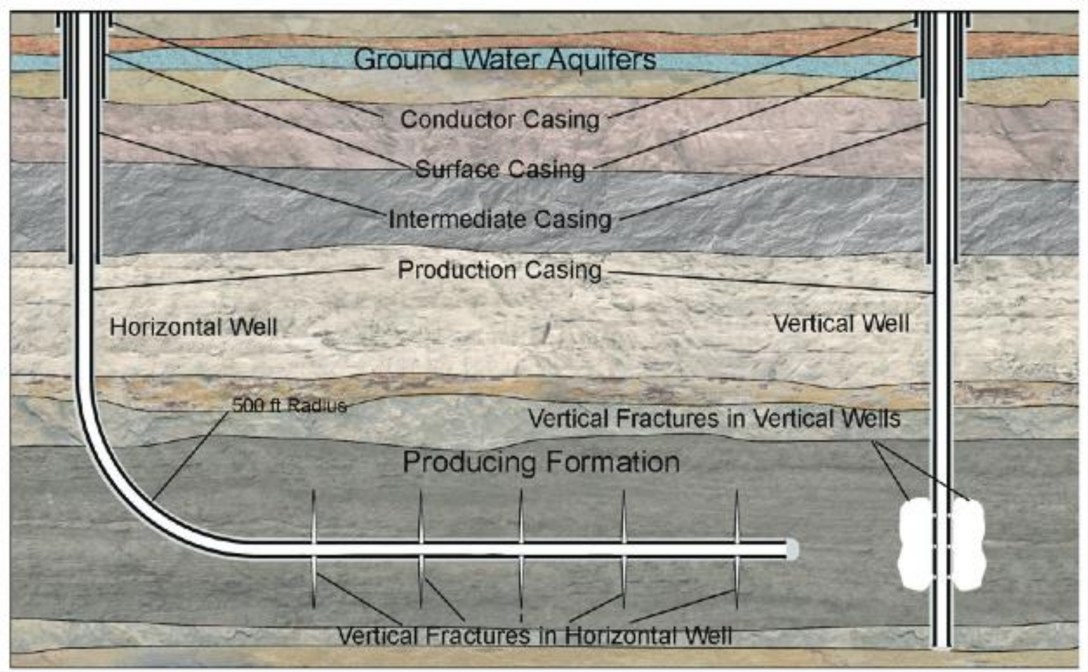
With my knowledge in advanced statistical algorithms, **machine learning** and forecasting, I strive to bring innovative, highly efficient, and high-quality technical solutions to businesses.

linkedin.com/albertompalaciobastos   github.com/albertompalaciobastos

Exercise Overview

S&P Global
Commodity Insights

Shale gas well



Problem

As stated by the company executives, the objective is to build a model to estimate gas production of natural gas shale wells.

The **Business Goal** is to predict production of shale gas wells unseen by the model.

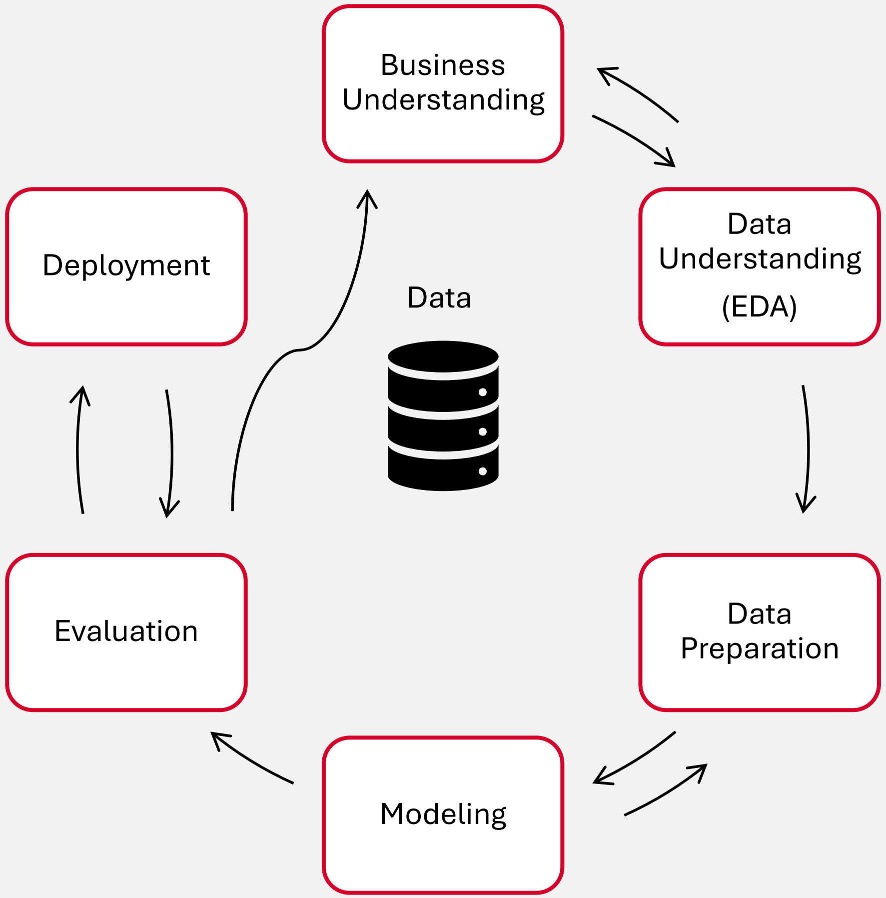
Objective: Build a model to estimate gas production.

Dataset: http://huy302.github.io/interview_dataset.csv

Analysis Attack Plan

S&P Global
Commodity Insights

CRISP-DM



1. Business Understanding.

1.1 Understand the question and business needs.

Develop a predictive analysis based on a machine learning algorithm to estimate/forecast the gas production of shale gas wells.

1.2 Determine appropriate analytic approach.

Since the target is numerical continuous variable, this model requires a Regression model. Regression models to test:

- Linear Regression.
- Decision Tree Regressor.
- Random Forest Regressor.
- Gradient Boosting Regressor.
- Supervised Neural Network Regressor.

Analytics approach: Train a Regression Model

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 28 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   treatment_company       1000 non-null   object
 1   azimuth                 945 non-null    float64
 2   md                      1000 non-null   int64
 3   tvd                     980 non-null    float64
 4   date_on_production      1000 non-null   object
 5   operator                1000 non-null   object
 6   footage_lateral_length  1000 non-null   float64
 7   well_spacing            844 non-null    float64
 8   porpoise_deviation      1000 non-null   float64
 9   porpoise_count          1000 non-null   int64
 10  shale_footage           1000 non-null   int64
 11  acoustic_impedance      1000 non-null   float64
 12  log_permeability        1000 non-null   float64
 13  porosity                881 non-null    float64
 14  poisson_ratio           1000 non-null   float64
 15  water_saturation        423 non-null    float64
 16  toc                     979 non-null    float64
 17  vcl                     1000 non-null   float64
 18  p-velocity              1000 non-null   float64
 19  s-velocity              1000 non-null   float64
 20  youngs_modulus          981 non-null    float64
 21  isip                    923 non-null    float64
 22  breakdown_pressure      256 non-null    float64
 23  pump_rate               1000 non-null   int64
 24  total_number_of_stages  1000 non-null   int64
 25  proppant_volume         868 non-null    float64
 26  proppant_fluid_ratio    1000 non-null   float64
 27  production              1000 non-null   float64
dtypes: float64(20), int64(5), object(3)
memory usage: 218.9+ KB
```

- Dataset consists of 28 columns (features) and 1000 rows (records).
- Two categorical variables in the dataset: `treatment_company` and `operator`.
- One date type variable is identified: `date_on_production`.
- An `age` feature can be extracted by subtracting the `date_on_production` from todays date.
- There are no duplicates in the data.

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

```python
# Check for missing values
df1.isna().sum()
```

```
treatment_company          0
azimuth                   55
md                         0
tvd                       20
date_on_production         0
operator                   0
footage_lateral_length     0
well_spacing             156
porpoise_deviation         0
porpoise_count             0
shale_footage              0
acoustic_impedance         0
log_permeability           0
porosity                 119
poisson_ratio              0
water_saturation         577
toc                       21
vcl                        0
p-velocity                 0
s-velocity                 0
youngs_modulus            19
isip                      77
breakdown_pressure       744
pump_rate                  0
total_number_of_stages     0
proppant_volume          132
proppant_fluid_ratio       0
production                 0
year_on_production         0
age                        0
dtype: int64
```

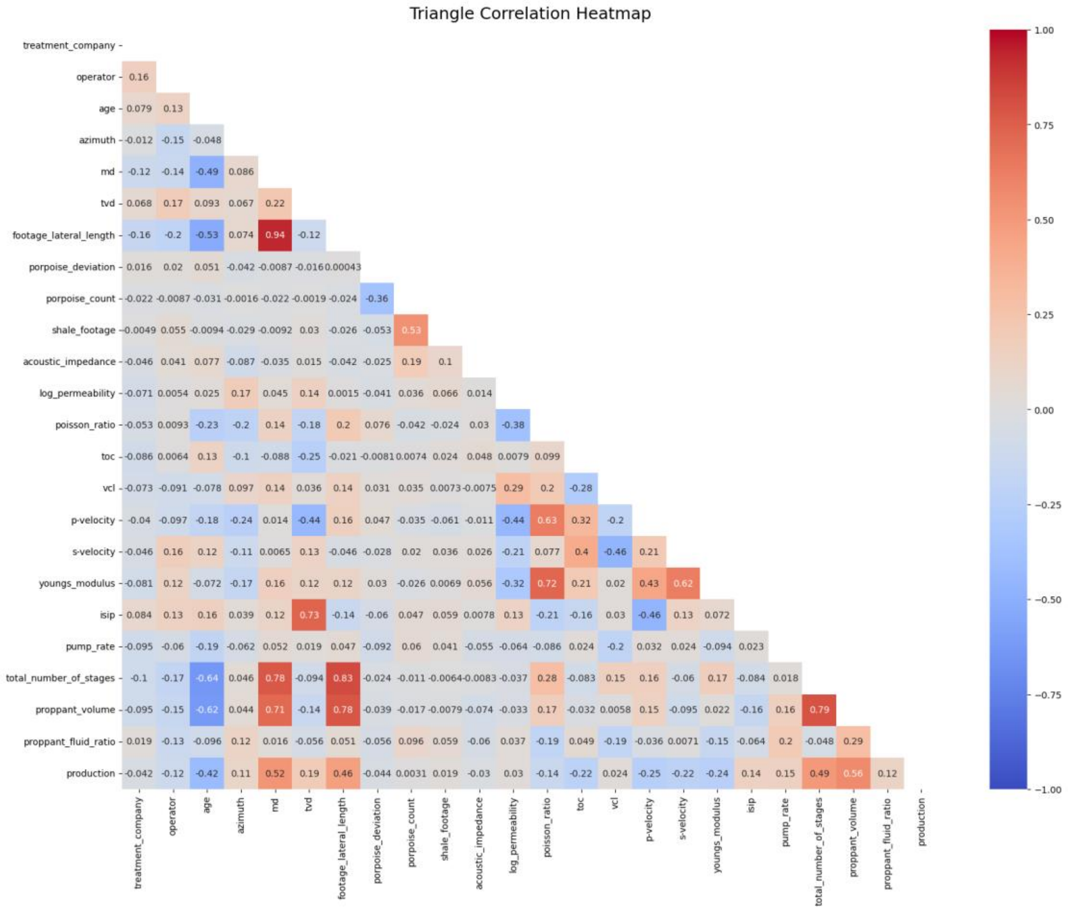
Feature	Null value percentage	Pearson Correlation Coefficient with target: `production`	Missing Values Strategy	Risk of information loss
`azimuth`	5.5%	0.13	Replace missing values with mean	Low
`tvd`	2.0%	0.18	Delete row / delete record	Low
`well_spacing`	15.6%	0.017	Delete column / delete feature	Low
`porosity`	11.9%	0.03	Delete column / delete feature	Low
`water_saturation`	57.7% (high)	0.05 (low)	Delete column / delete feature	Medium
`toc`	2.1%	-0.20	Replace missing values with mean	Low
`youngs_modulus`	1.9%	-0.24	Delete row / delete record	Medium
`isip`	7.7%	0.15	Replace missing values with mean	Low
`breakdown_pressure`	74.4% (high)	0.017 (low)	Delete column / delete feature	Low
`proppant_volume`	13.2%	0.57	Delete row / delete record	Medium

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 837 entries, 0 to 836
Data columns (total 24 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   treatment_company       837 non-null    int64
 1   operator                837 non-null    int64
 2   age                     837 non-null    int64
 3   azimuth                 837 non-null    float64
 4   md                      837 non-null    int64
 5   tvd                     837 non-null    float64
 6   footage_lateral_length  837 non-null    float64
 7   porpoise_deviation      837 non-null    float64
 8   porpoise_count          837 non-null    int64
 9   shale_footage           837 non-null    int64
 10  acoustic_impedance      837 non-null    float64
 11  log_permeability        837 non-null    float64
 12  poisson_ratio           837 non-null    float64
 13  toc                     837 non-null    float64
 14  vcl                     837 non-null    float64
 15  p-velocity              837 non-null    float64
 16  s-velocity              837 non-null    float64
 17  youngs_modulus          837 non-null    float64
 18  isip                    837 non-null    float64
 19  pump_rate               837 non-null    int64
 20  total_number_of_stages  837 non-null    int64
 21  proppant_volume         837 non-null    float64
 22  proppant_fluid_ratio    837 non-null    float64
 23  production              837 non-null    float64
dtypes: float64(16), int64(8)
memory usage: 157.1 KB
```

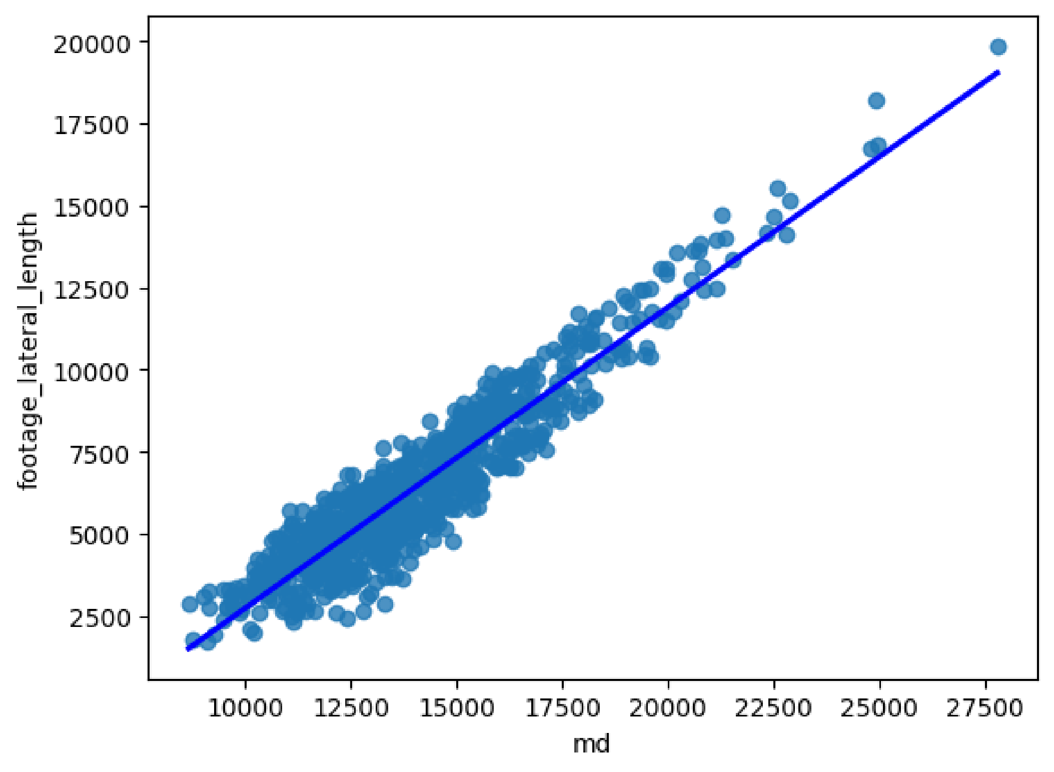


Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

- There is a high positive correlation between `md` and `footage_lateral_length`: 0.94.
- This is due to that `md` is the measured depth of the well in feet and it includes the horizontal well section, which is the definition of `footage_lateral_length`.
- `md` has a higher correlation with the dependent variable `production` (0.52), than **`footage_lateral_length`** (0.46), the latter **should be dropped** from the features feed to the model.



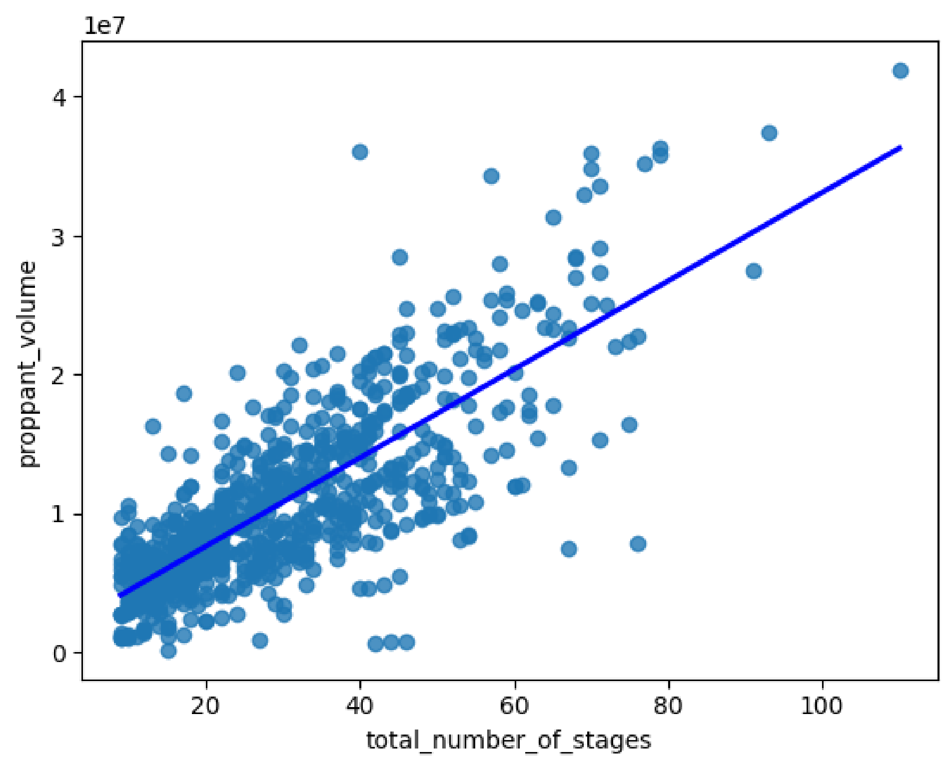
Analytics approach: Train a Regression Model

Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

- High positive correlation between `proppant_volume` and `total_number_of_stages`: 0.79.

- This is due to that `proppant_volume` is the amount of proppant fracturing fluid used in the completion of a well (lbs), and it is correlated to the total stages used to fracture the horizontal section of the well, which is the definition of `total_number_of_stages`.

- `proppant_volume` has a higher correlation with the dependent variable `production` (0.56) than **`total_number_of_stages`** (0.49), the latter **should be dropped** from the features feed to the model.



S&P Global
Commodity Insights

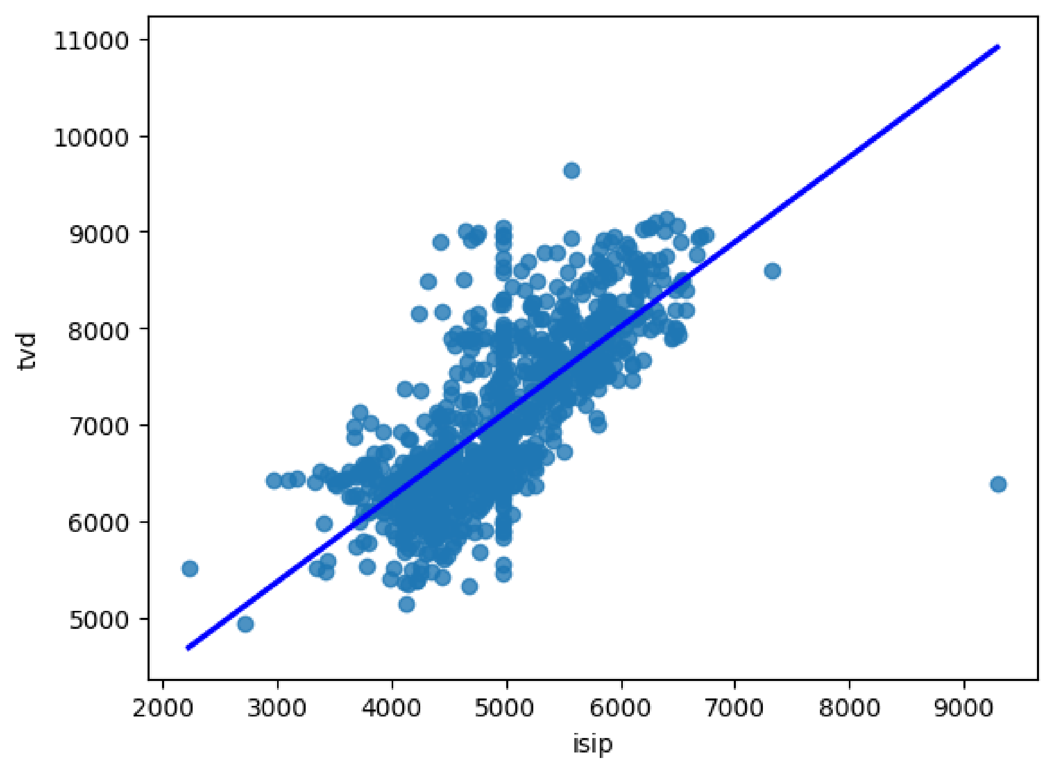
Analytics approach: Train a Regression Model

Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

- There is a high positive correlation between \`isip\` and \`tvd\`: 0.73.

- This is due to that \`isip\` is the instantaneous shut-in pressure (when the pumps are quickly stopped, and the fluids stop moving, the friction pressures disappear and its main component is the static hidraulic pressure), and it is correlated to the true vertical depth, wich is the definition of \`tvd\`.

- \`tvd\` has a higher correlation with the dependent variable \`production\` (0.19), than **\`isip\`** (0.14), the latter **should be dropped** from the features feed to the model.



Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

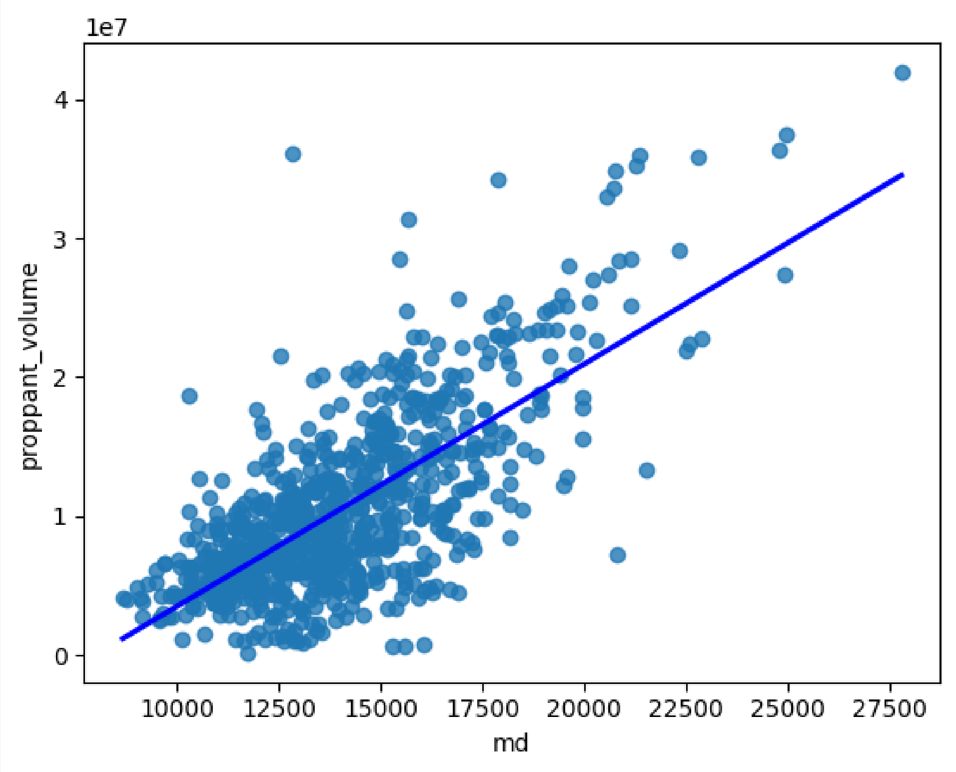
Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

- There is a high positive correlation between \`md\` and \`proppant_volume\`: 0.71.

- \`proppant_volume\` is highly correlated with the target variable \`production\` (0.56), a **new feature** will be created as a unitary measure of proppant fluid used in the completion of the well.

$$Eq. (1)$$

$$\text{unit propant volume} = \frac{\text{proppant volume}}{\text{md}}$$



Analytics approach: Train a Regression Model

S&P Global Commodity Insights

Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

- There is a high positive correlation between \`youngs_modulus\` and \`poisson_ratio\`: 0.72.
- Based on fracture mechanics, a more brittle formation is easier to fracture [1]. As stated by Rickman et al. [2] a empirical correlation such as Young's modulus v. Poisson ratio is convenient to use as a brittleness index to assist in locating the preferred injection intervals.
- Based on the laboratory ultrasonic measurements to derive the relationship between dynamic Young's modulus and Poisson's ratio, Rickman et al. [3] proposed the following equation to evaluate the brittleness:

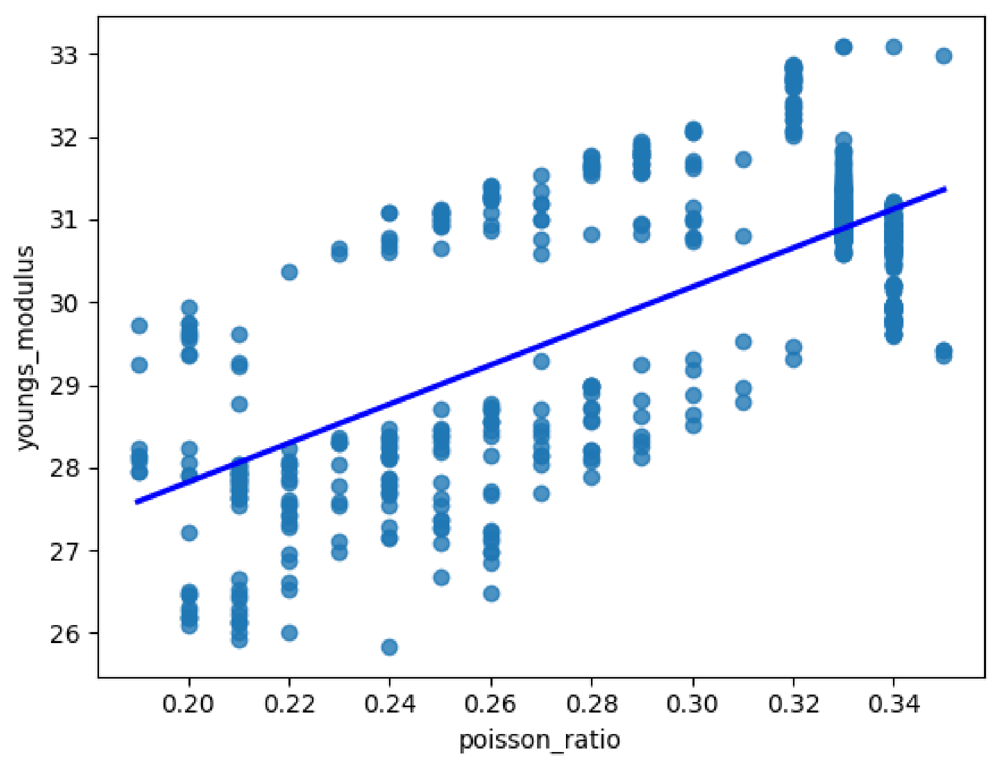
$$Eq.(2)$$

$$Br = (50/7)(E - 28v + 10.2)$$

Br = Brittleness ratio

E = Young's modulus

v = Poisson's ratio



[1] A.T. Zehnder, Fracture mechanics, in: Lecture Notes in Applied and Computational Mechanics, vol. 62, Springer Sci. & Business Media, 2012.

[2] R. Rickman, M. Mullen, E. Petre, B. Grieser, D. Kundert, A practical use of shale petrophysics for stimulation design optimization: all shale plays are not clones of the Barnett shale, in: SPE 115258, SPE ATCE, Denver, CO, USA, Sept. 21e24, 2008.

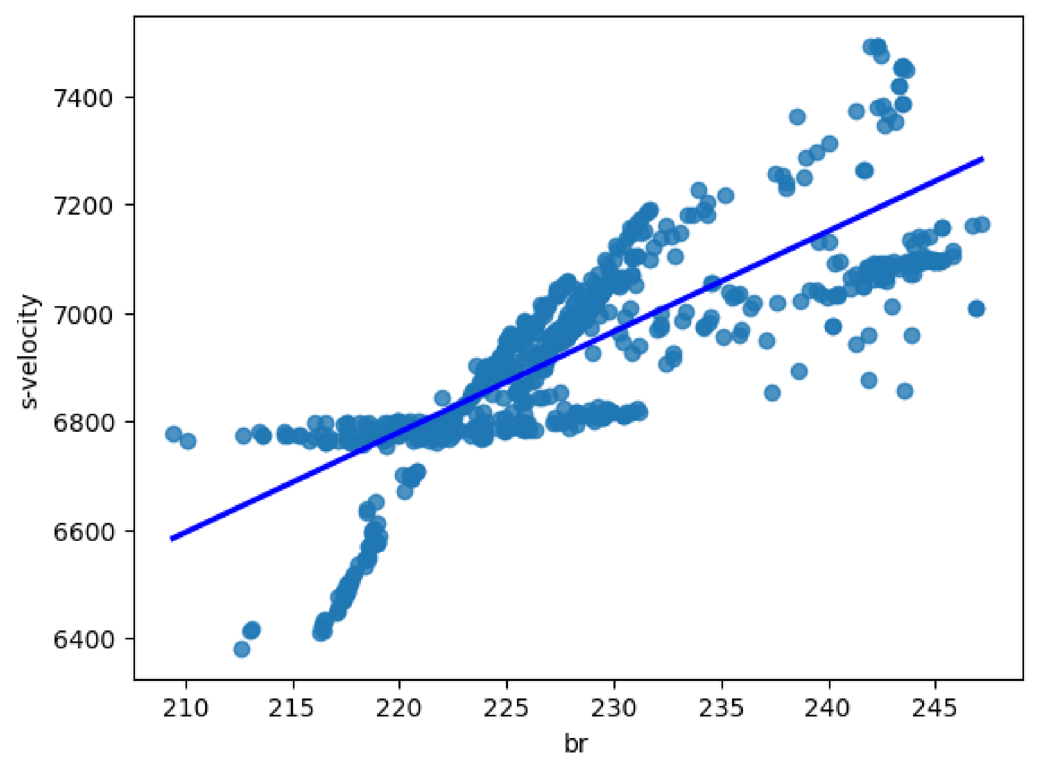
[3] Mao Bai, Why are brittleness and fracability not equivalent in designing hydraulic fracturing in tight shale gas reservoirs, Petroleum, Volume 2, Issue 1, 2016, Pages 1-19, ISSN 2405-6561, https://doi.org/10.1016/j.petlm.2016.01.001.

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

- There is a high positive correlation between `br` and `s-velocity`: 0.78.
- The scatter plot shows certain structure between 'br' and 's-velocity', this is not recommended for regression model training. This could be considered as evidence of different types of shale rock or other well properties.
- It is known that the Young's modulus (component of `br`) is correlated with the shear sound wave velocity (`s-velocity`) and is an indicator of rock porosity in shale deposits [4].
- Considering that the **shear wave velocity** is used to estimate the Young's modulus and other well properties that are already in the dataset, it **should be dropped**.



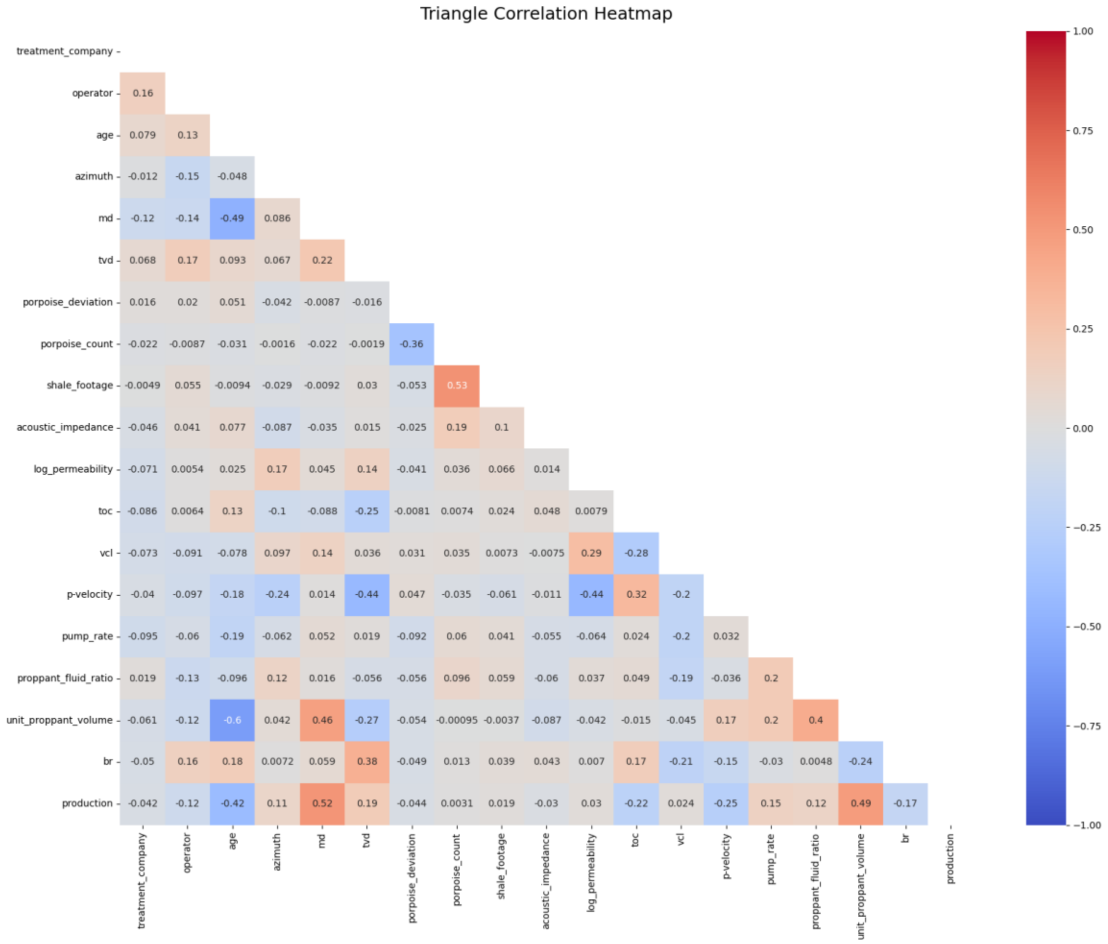
[4] Sharifi, J., Nooraiepour, M., Amiri, M. et al. Developing a relationship between static Young's modulus and seismic parameters. J Petrol Explor Prod Technol 13, 203–218 (2023). https://doi.org/10.1007/s13202-022-01546-6

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → **Data Cleaning** → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → Deployment

```
<class 'pandas.core.frame.DataFrame'>
Index: 558 entries, 0 to 836
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   treatment_company     558 non-null    int64
 1   operator              558 non-null    int64
 2   age                   558 non-null    int64
 3   azimuth               558 non-null    float64
 4   md                    558 non-null    int64
 5   tvd                   558 non-null    float64
 6   porpoise_deviation    558 non-null    float64
 7   porpoise_count        558 non-null    int64
 8   shale_footage         558 non-null    int64
 9   acoustic_impedance    558 non-null    float64
 10  log_permeability      558 non-null    float64
 11  toc                   558 non-null    float64
 12  vcl                   558 non-null    float64
 13  p-velocity            558 non-null    float64
 14  pump_rate             558 non-null    int64
 15  proppant_fluid_ratio  558 non-null    float64
 16  unit_proppant_volume  558 non-null    float64
 17  br                    558 non-null    float64
 18  production            558 non-null    float64
dtypes: float64(12), int64(7)
memory usage: 87.2 KB
```

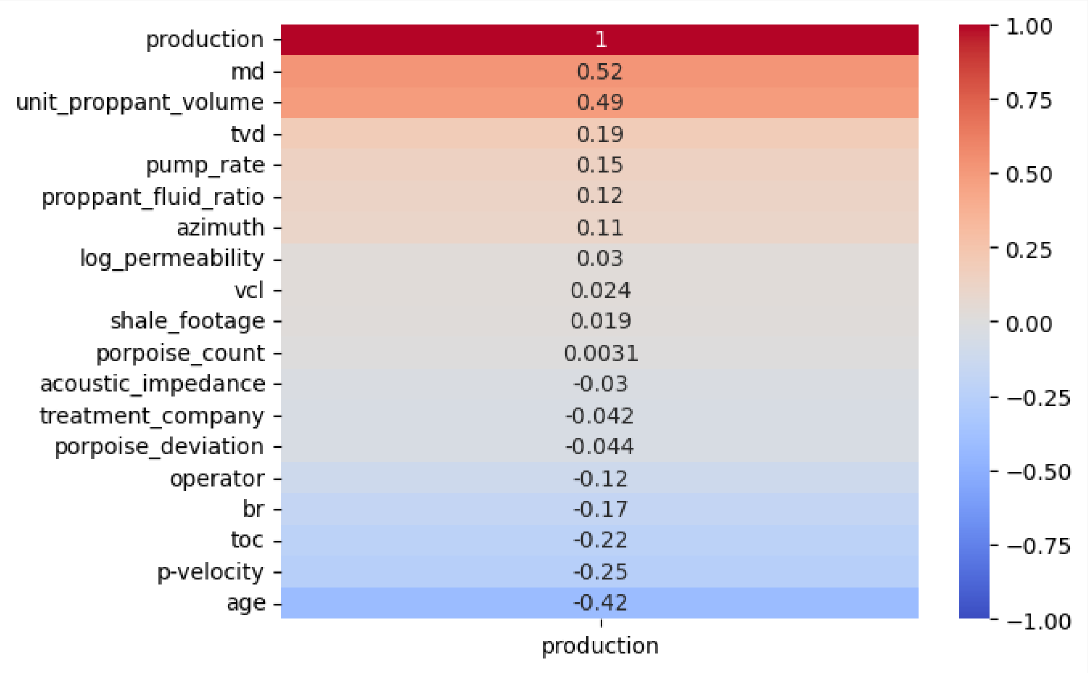


Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → **Exploratory Data Analysis (EDA)** → Data Preparation → Model Training → Evaluation → Deployment

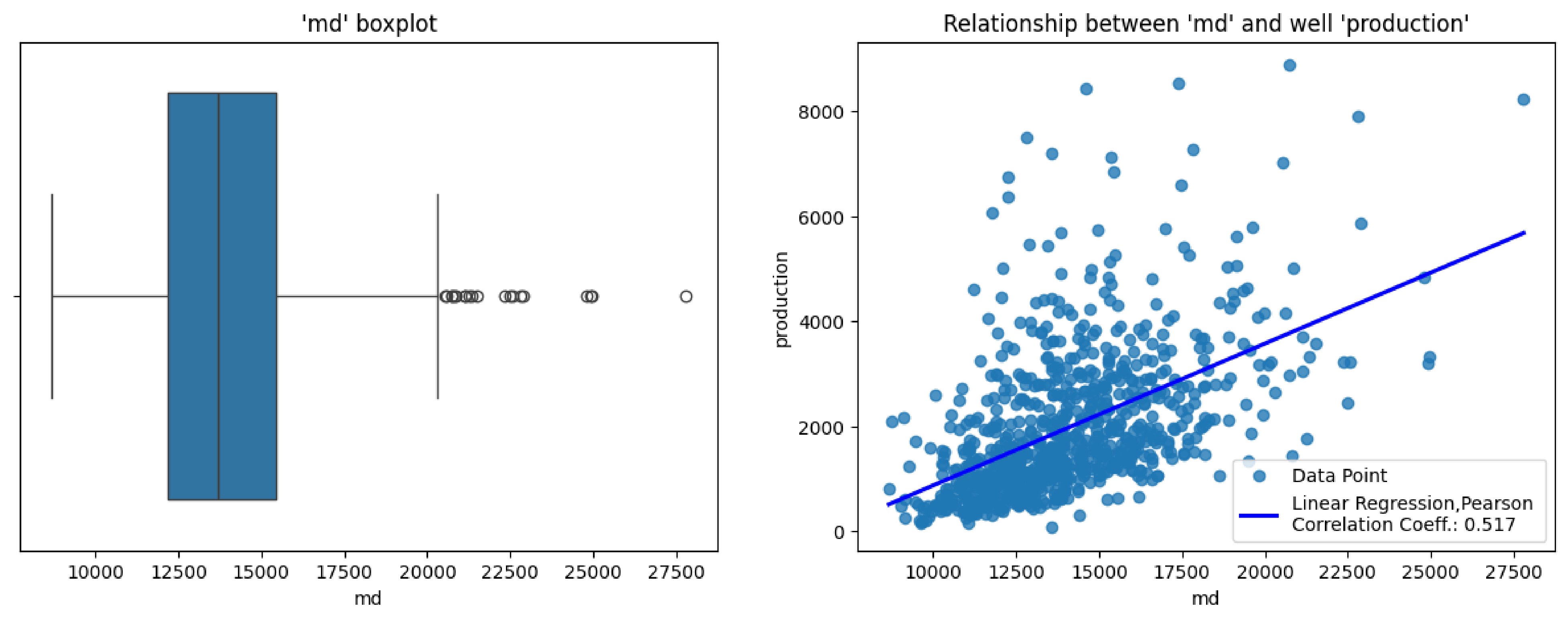
```
<class 'pandas.core.frame.DataFrame'>
Index: 558 entries, 0 to 836
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   treatment_company     558 non-null    int64
 1   operator              558 non-null    int64
 2   age                   558 non-null    int64
 3   azimuth               558 non-null    float64
 4   md                    558 non-null    int64
 5   tvd                   558 non-null    float64
 6   porpoise_deviation    558 non-null    float64
 7   porpoise_count        558 non-null    int64
 8   shale_footage         558 non-null    int64
 9   acoustic_impedance    558 non-null    float64
 10  log_permeability      558 non-null    float64
 11  toc                   558 non-null    float64
 12  vcl                   558 non-null    float64
 13  p-velocity            558 non-null    float64
 14  pump_rate             558 non-null    int64
 15  proppant_fluid_ratio  558 non-null    float64
 16  unit_proppant_volume  558 non-null    float64
 17  br                    558 non-null    float64
 18  production            558 non-null    float64
dtypes: float64(12), int64(7)
memory usage: 87.2 KB
```



Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

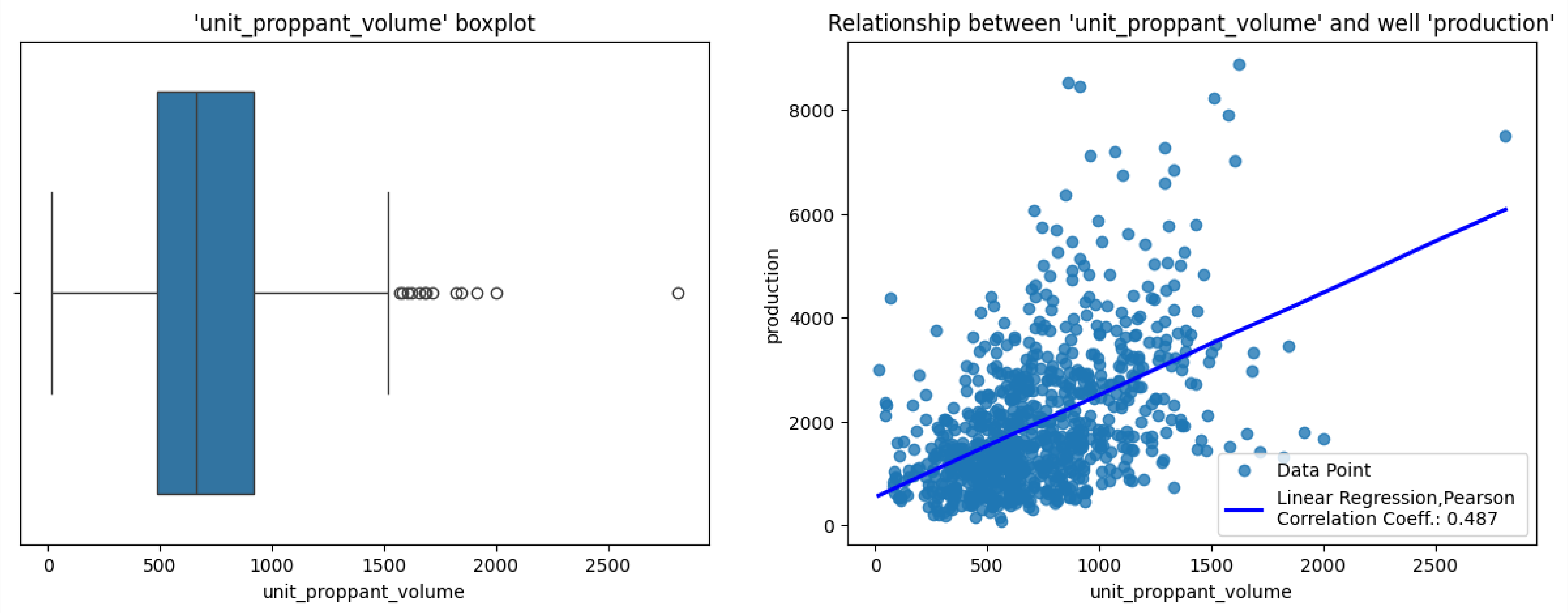
Data Collection → Data Cleaning → **Exploratory Data Analysis (EDA)** → Data Preparation → Model Training → Evaluation → Deployment



The longer the well, the more productive it is

Analytics approach: Train a Regression Model

Data Collection → Data Cleaning → **Exploratory Data Analysis (EDA)** → Data Preparation → Model Training → Evaluation → Deployment

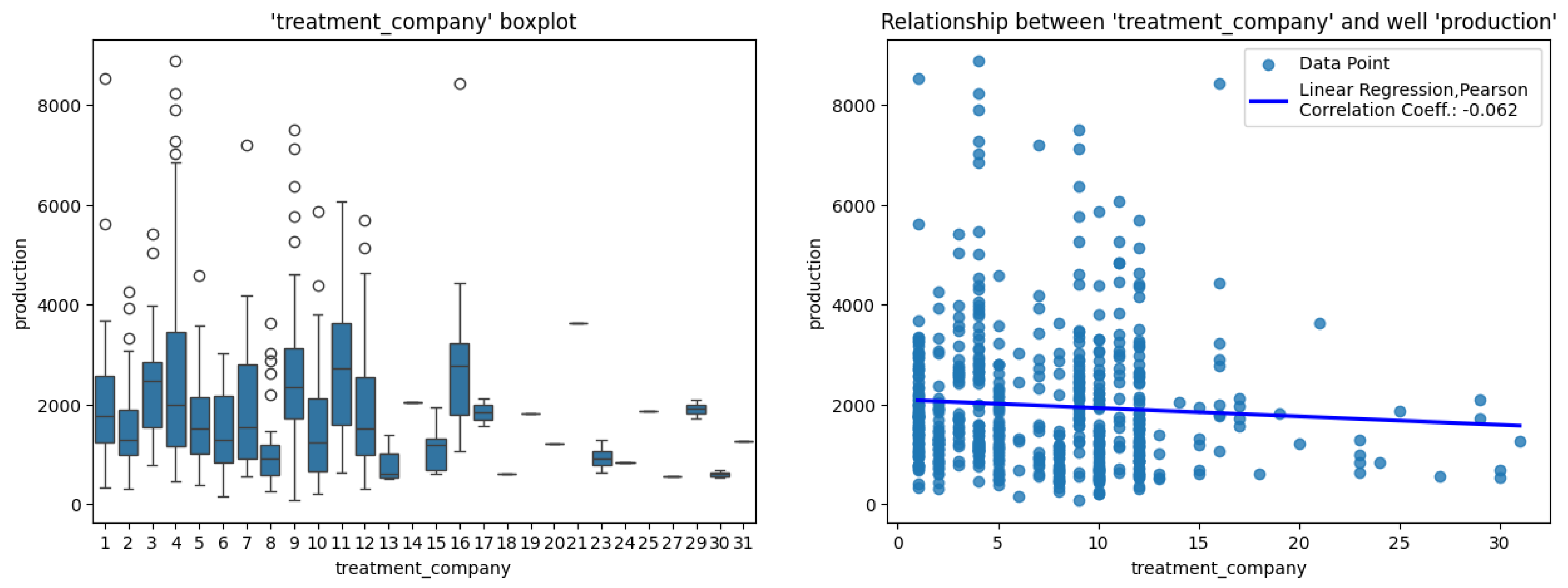


With more proppant fluid used in well completion, the more productive it is

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → **Exploratory Data Analysis (EDA)** → Data Preparation → Model Training → Evaluation → Deployment

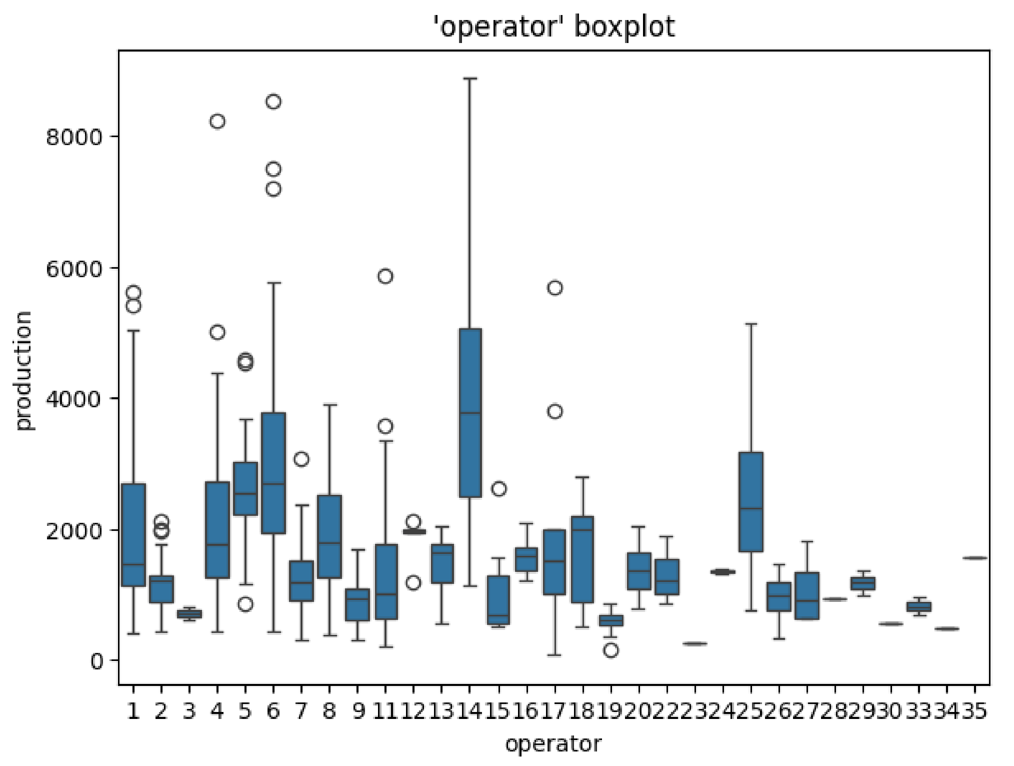


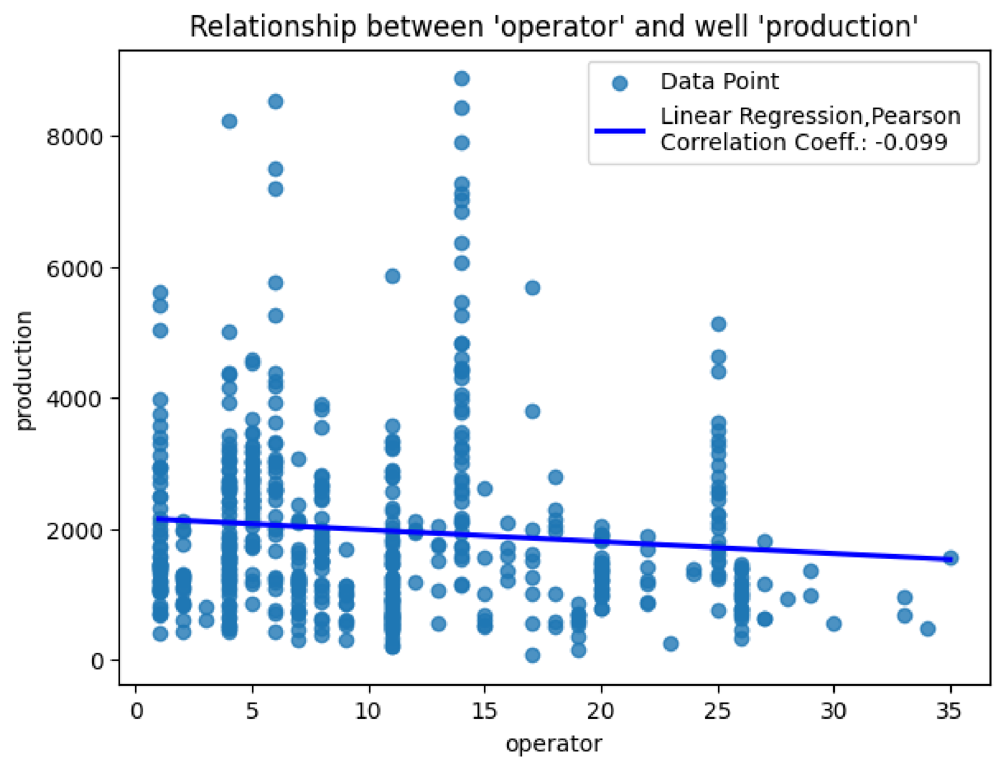
There are some treatment companies with significant differences in well production

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → **Exploratory Data Analysis (EDA)** → Data Preparation → Model Training → Evaluation → Deployment



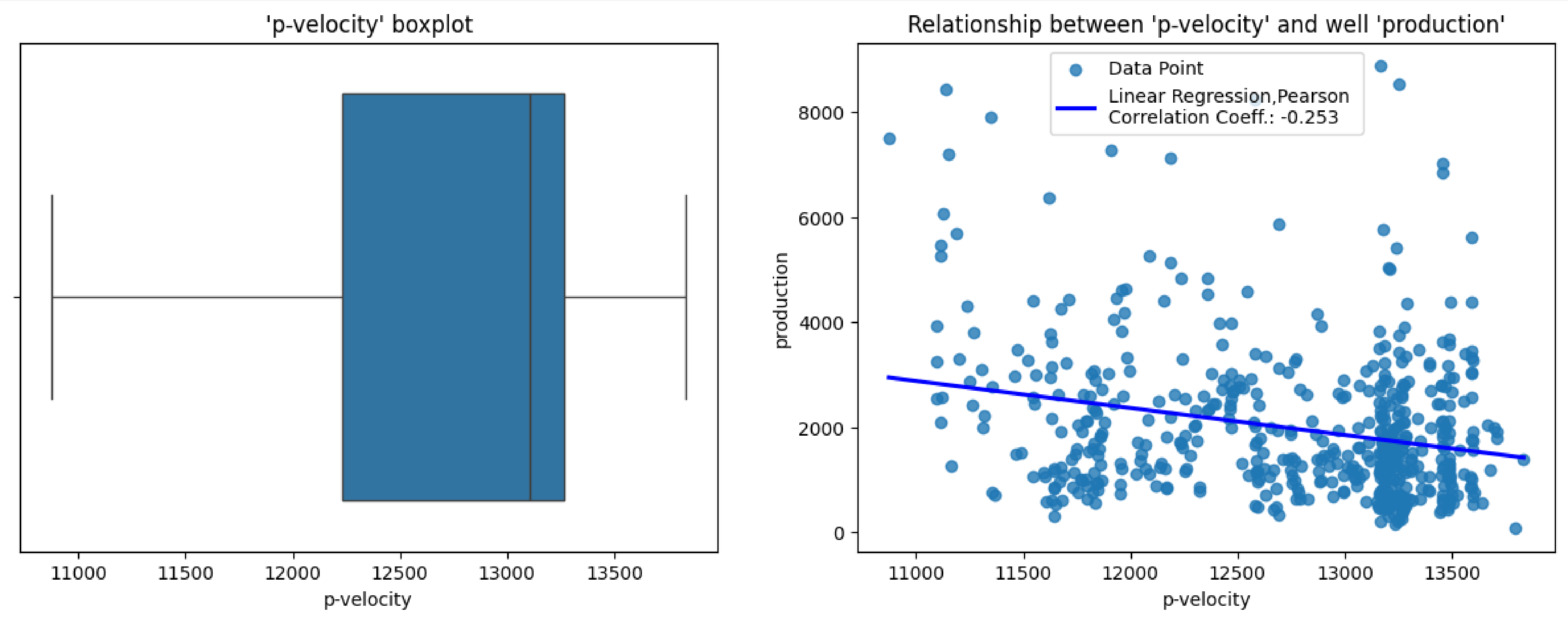


There are some operator companies with significant differences in well production

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → **Exploratory Data Analysis (EDA)** → Data Preparation → Model Training → Evaluation → Deployment

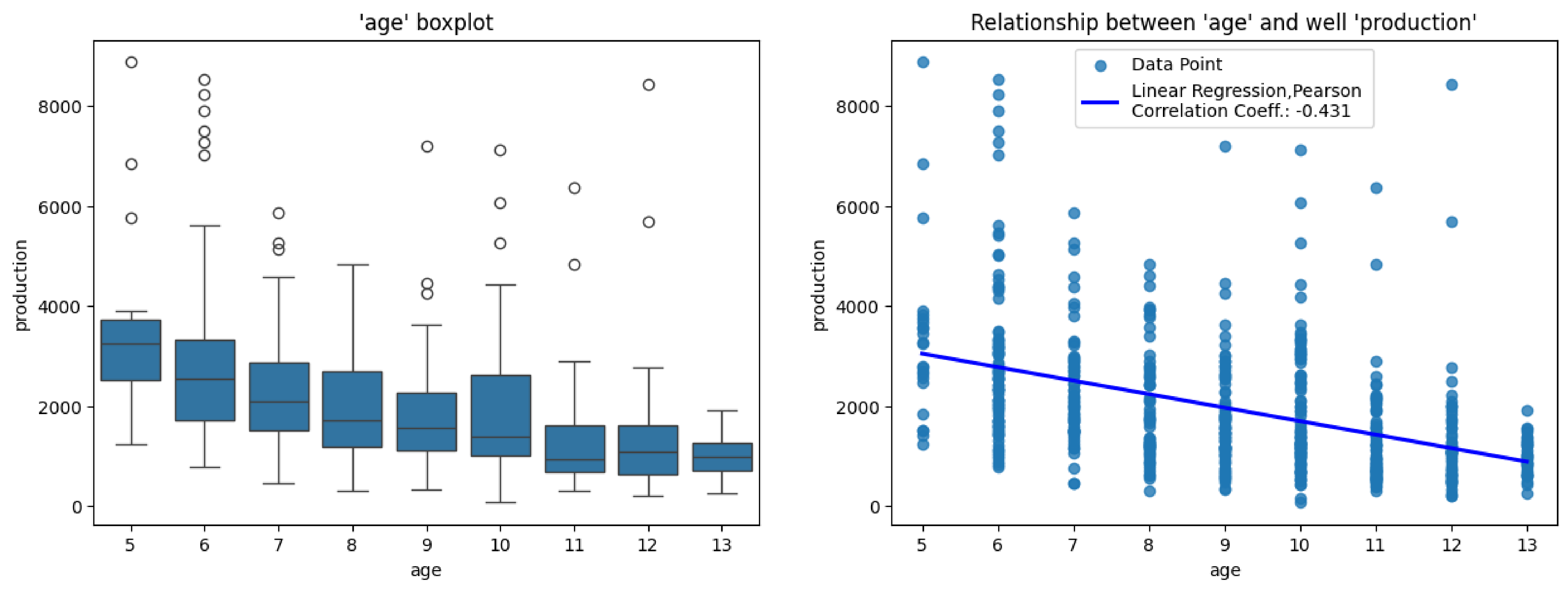


The higher the compression wave velocity, the less productive the well.

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → **Exploratory Data Analysis (EDA)** → Data Preparation → Model Training → Evaluation → Deployment

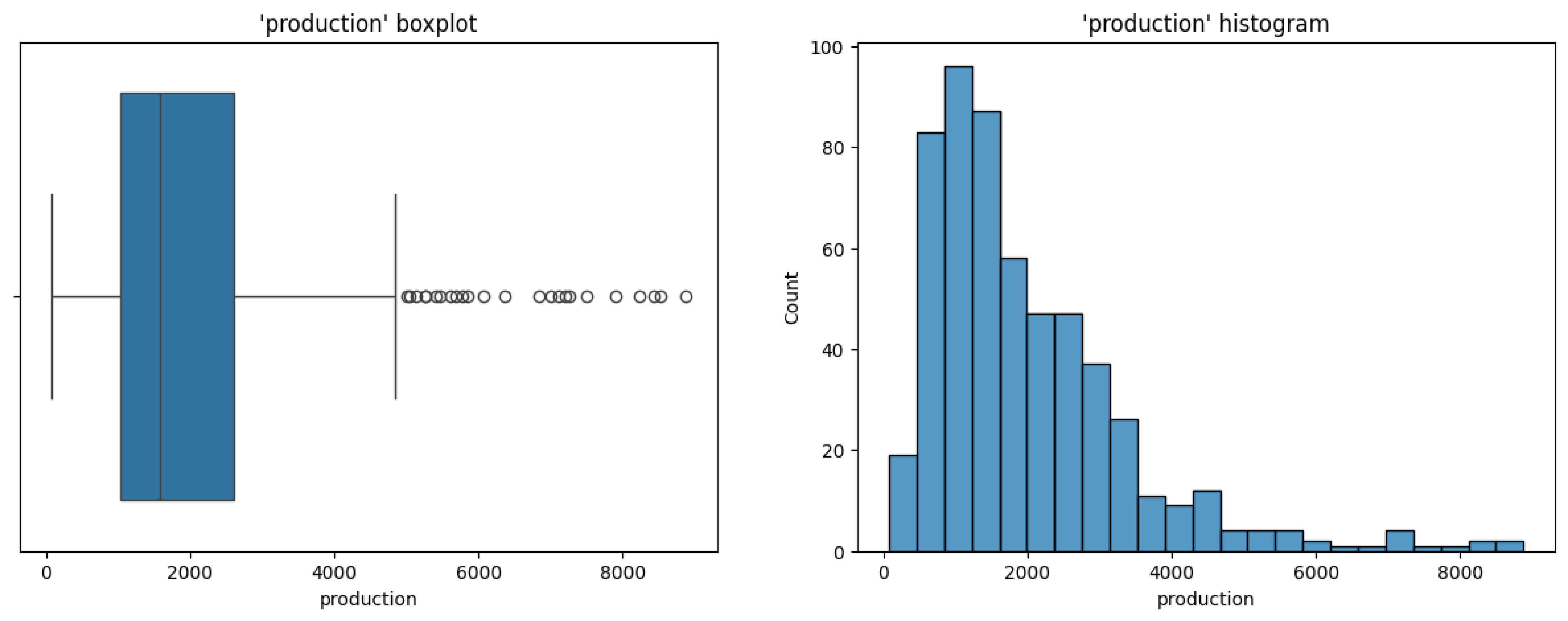


The older the well, the less productive it is.

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → **Exploratory Data Analysis (EDA)** → Data Preparation → Model Training → Evaluation → Deployment



There are some outliers in production. They are dropped from the training dataset to avoid model bias.

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → **Data Preparation** → Model Training → Evaluation → Deployment

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 534 entries, 0 to 533
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   treatment_company      534 non-null    int64
 1   operator               534 non-null    int64
 2   age                    534 non-null    int64
 3   azimuth                534 non-null    float64
 4   md                     534 non-null    int64
 5   tvd                    534 non-null    float64
 6   porpoise_deviation     534 non-null    float64
 7   porpoise_count         534 non-null    int64
 8   shale_footage          534 non-null    int64
 9   acoustic_impedance     534 non-null    float64
 10  log_permeability       534 non-null    float64
 11  toc                    534 non-null    float64
 12  vcl                    534 non-null    float64
 13  p-velocity             534 non-null    float64
 14  pump_rate              534 non-null    int64
 15  proppant_fluid_ratio   534 non-null    float64
 16  unit_proppant_volume   534 non-null    float64
 17  br                     534 non-null    float64
 18  production             534 non-null    float64
dtypes: float64(12), int64(7)
memory usage: 79.4 KB
```

- Isolate features and target.

```
Features shape: (534, 18)
 Target shape: (534,)
```

- Encode categorical variables `treatment_company` and `operator`

```
RangeIndex: 534 entries, 0 to 533
Data columns (total 34 columns):
```

- Scale features: MaxAbsScaler.

	age	azimuth	md	tvd	porpoise_deviation	porpoise_count	shale_footage
0	0.461538	-0.322844	0.661973	0.651483	0.002073	0.153846	0.183190
1	0.923077	-0.589719	0.491410	0.721076	0.006126	0.589744	0.420537

Analytics approach: Train a Regression Model

S&P Global Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → **Data Preparation** → Model Training → Evaluation → Deployment

Rank most important features:

	feature	ranking
15	br	1
11	p-velocity	2
2	md	3
3	tvd	4
14	unit_proppant_volume	5
32	operator_14	6
0	age	7
28	operator_6	8
9	toc	9
20	treatment_company_8	10
12	pump_rate	11
7	acoustic_impedance	12
29	operator_7	13
33	operator_infrequent_sklearn	14
31	operator_11	15
30	operator_8	16
26	operator_4	17

10	vcl	18
27	operator_5	19
23	treatment_company_12	20
6	shale_footage	21
21	treatment_company_9	22
17	treatment_company_2	23
18	treatment_company_4	24
19	treatment_company_5	25
13	proppant_fluid_ratio	26
16	treatment_company_1	27
25	operator_1	28
1	azimuth	29
4	porpoise_deviation	30
5	porpoise_count	31
22	treatment_company_10	32
8	log_permeability	33
24	treatment_company_infrequent_sklearn	34

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → **Data Preparation** → Model Training → Evaluation → Deployment

Split data into train and test datasets:

Encoded Unscaled dataset:

```
In [12]:
# Split data 70%-30% into training set and test set
X_train, X_test, y_train, y_test = train_test_split(encoded_features_df, target_df, test_size=0.30, random_state=22)
print ('Training Set: %d rows\nTest Set: %d rows' % (X_train.shape[0], X_test.shape[0]))

Training Set: 373 rows
Test Set: 161 rows
```

Encoded scaled dataset:

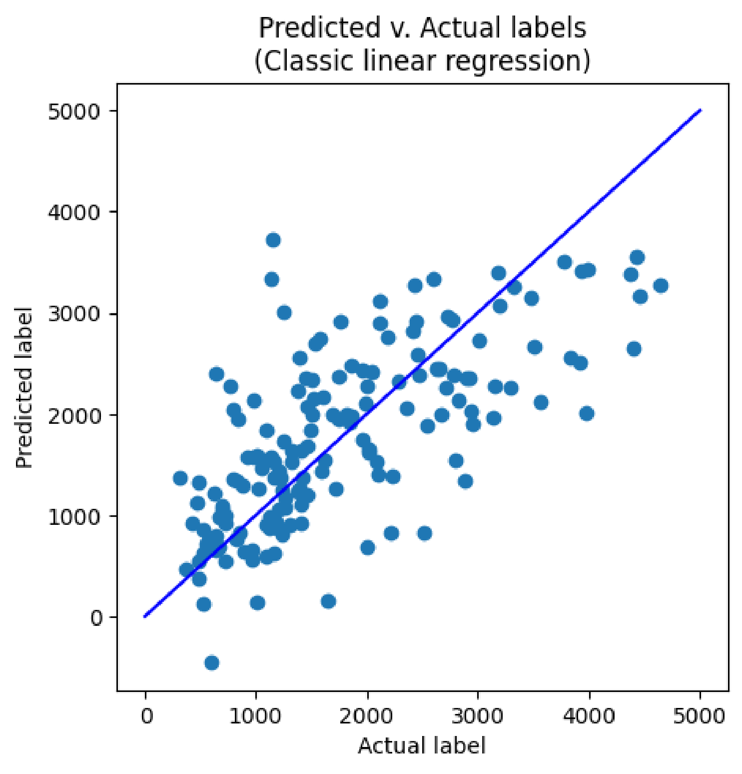
```
# Split data 70%-30% into training set and test set
X_train, X_test, y_train, y_test = train_test_split(scaled_features_df, target_df, test_size=0.30, random_state=22)
print ('Training Set: %d rows\nTest Set: %d rows' % (X_train.shape[0], X_test.shape[0]))

Training Set: 373 rows
Test Set: 161 rows
```

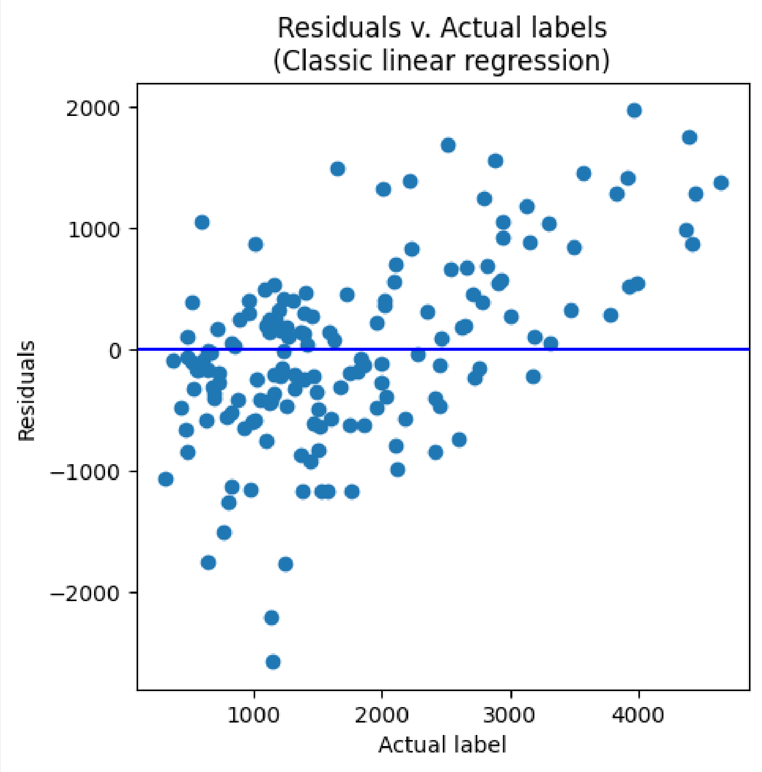
Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → **Model Training** → Evaluation → Deployment

Classic Multiple Linear Regressor:



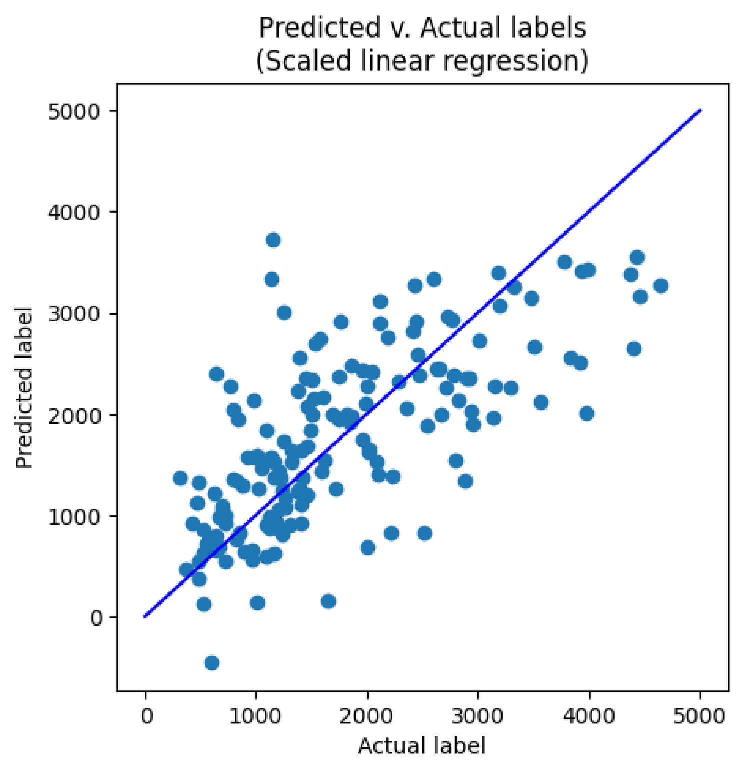


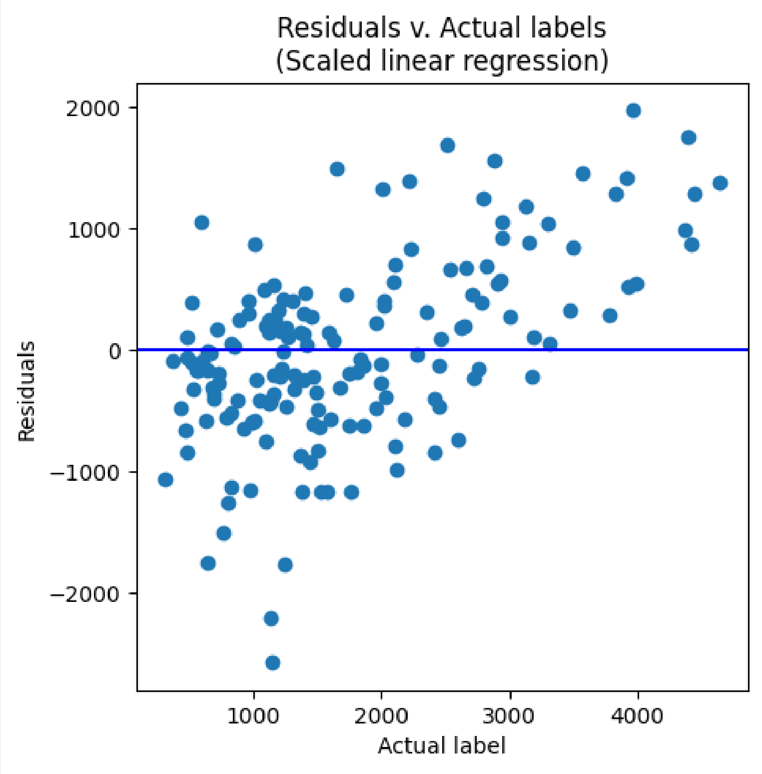
Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → **Model Training** → Evaluation → Deployment

Scaled Multiple Linear Regressor:



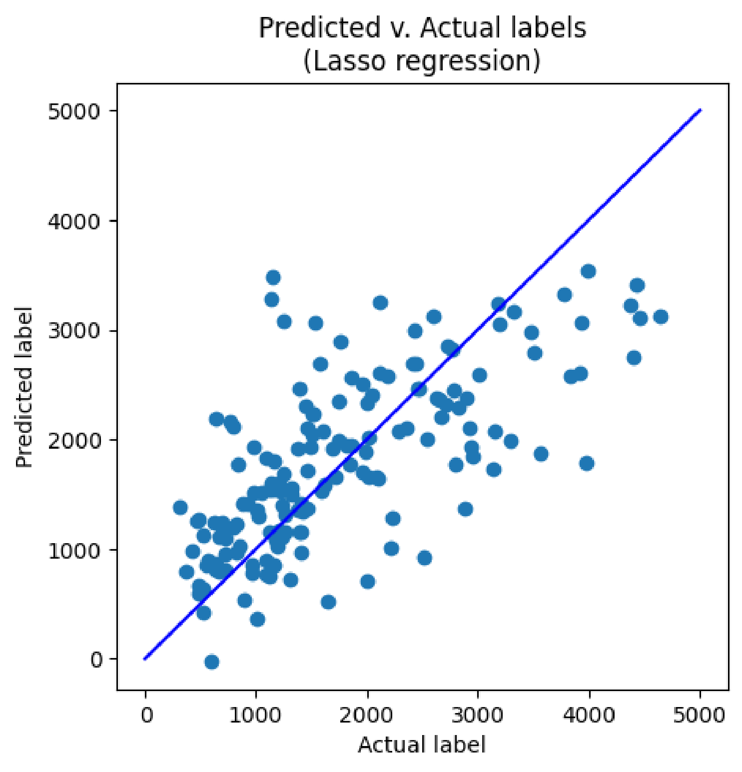


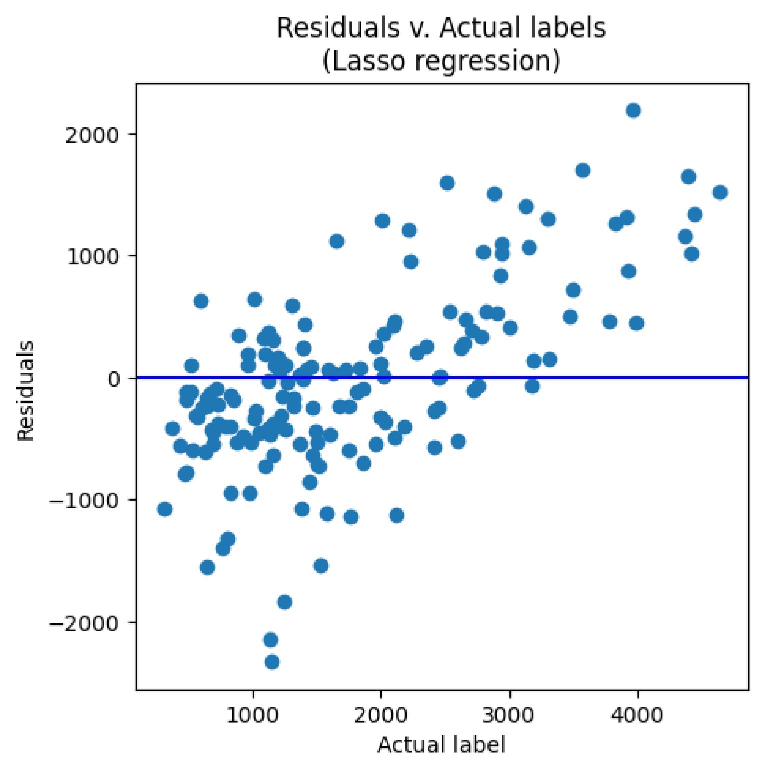
Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → **Model Training** → Evaluation → Deployment

Lasso Linear Regressor:



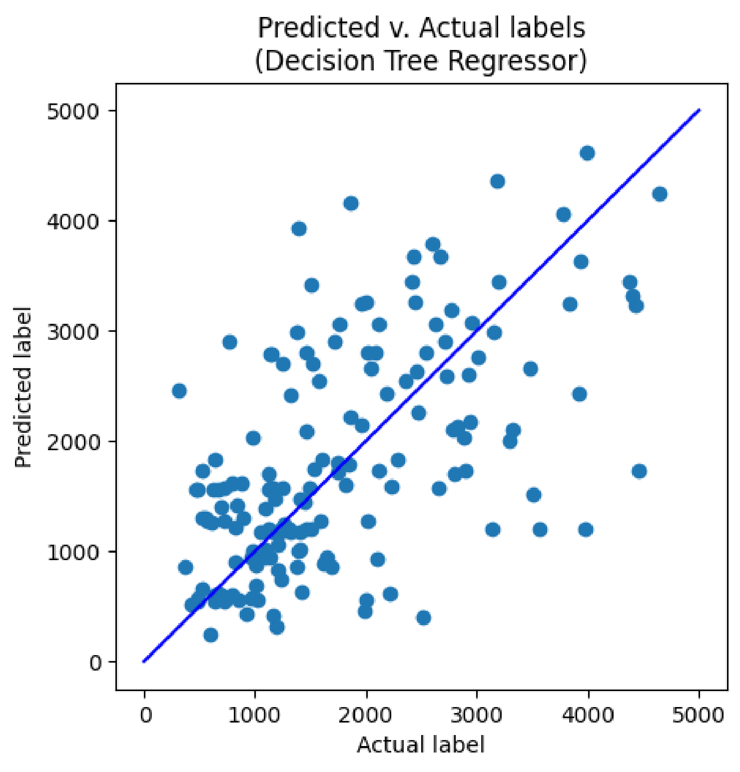


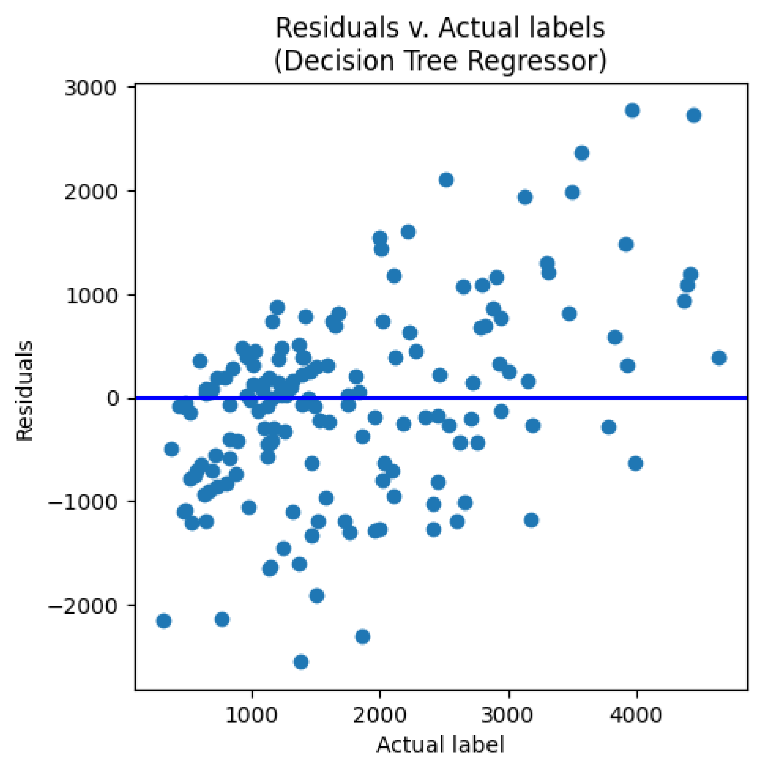
Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → **Model Training** → Evaluation → Deployment

Decision Tree Regressor:



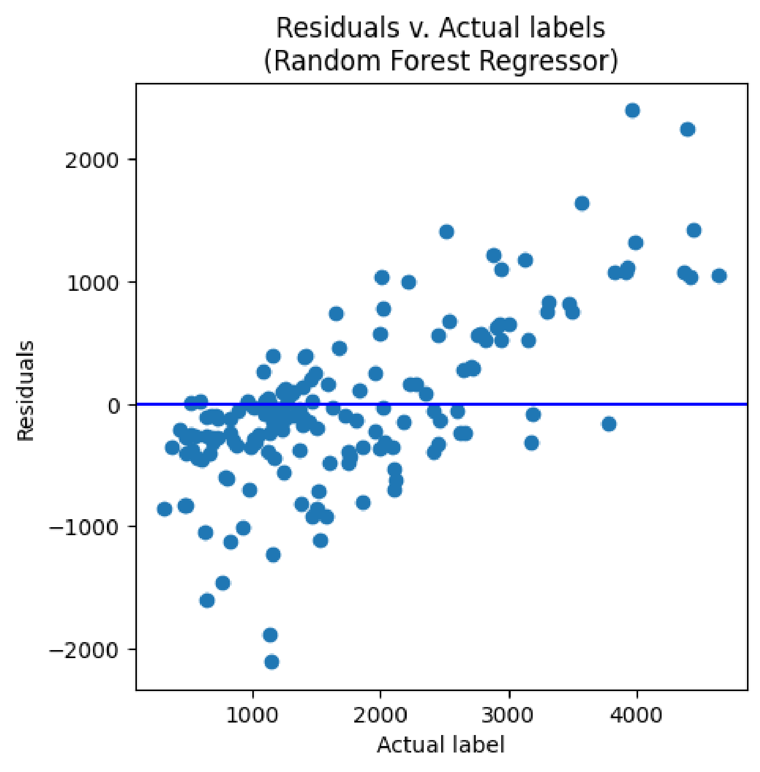


Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → **Model Training** → Evaluation → Deployment

Random Forest Regressor:

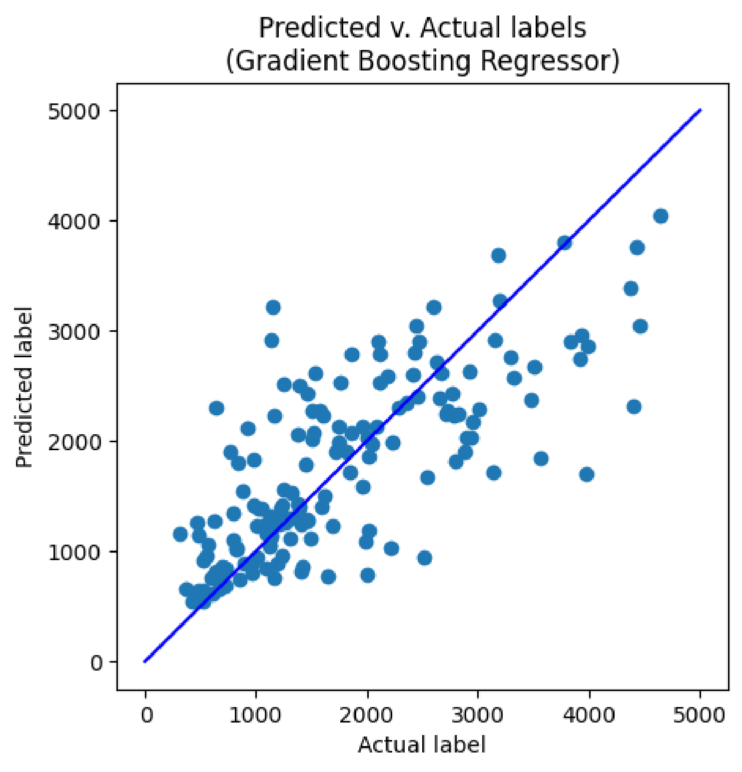


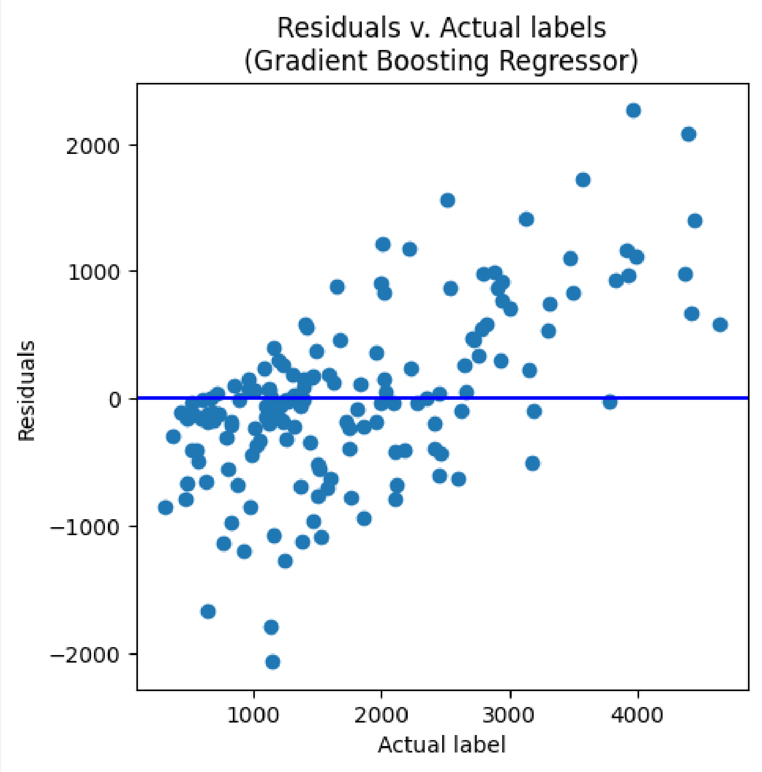
Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → **Model Training** → Evaluation → Deployment

Gradient Boosting Regressor:





Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → **Model Training** → Evaluation → Deployment

Neural Network Regressor:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 136)	4,760
dense_1 (Dense)	(None, 102)	13,974
dense_2 (Dense)	(None, 68)	7,004
dense_3 (Dense)	(None, 34)	2,346
dense_4 (Dense)	(None, 1)	35

Total params: 28,119 (109.84 KB)
Trainable params: 28,119 (109.84 KB)
Non-trainable params: 0 (0.00 B)

```
Epoch 1/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - loss: 1850.5558 - mse: 4546307.0000 - r2_score: -3.1184
Epoch 2/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1719.0503 - mse: 3978570.0000 - r2_score: -2.8876
Epoch 3/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1624.3894 - mse: 3727380.7500 - r2_score: -2.4495
Epoch 4/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1138.2734 - mse: 2141625.2500 - r2_score: -1.2208
Epoch 5/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 861.0742 - mse: 1122059.8750 - r2_score: -0.0296
Epoch 6/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 791.9641 - mse: 975193.4375 - r2_score: 0.0372
Epoch 7/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 743.6614 - mse: 958092.5625 - r2_score: 0.0098
Epoch 8/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 773.2810 - mse: 976790.3750 - r2_score: 0.1184
Epoch 9/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 701.9855 - mse: 849118.8125 - r2_score: 0.0861
Epoch 10/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 705.2977 - mse: 840860.7500 - r2_score: 0.1630
```

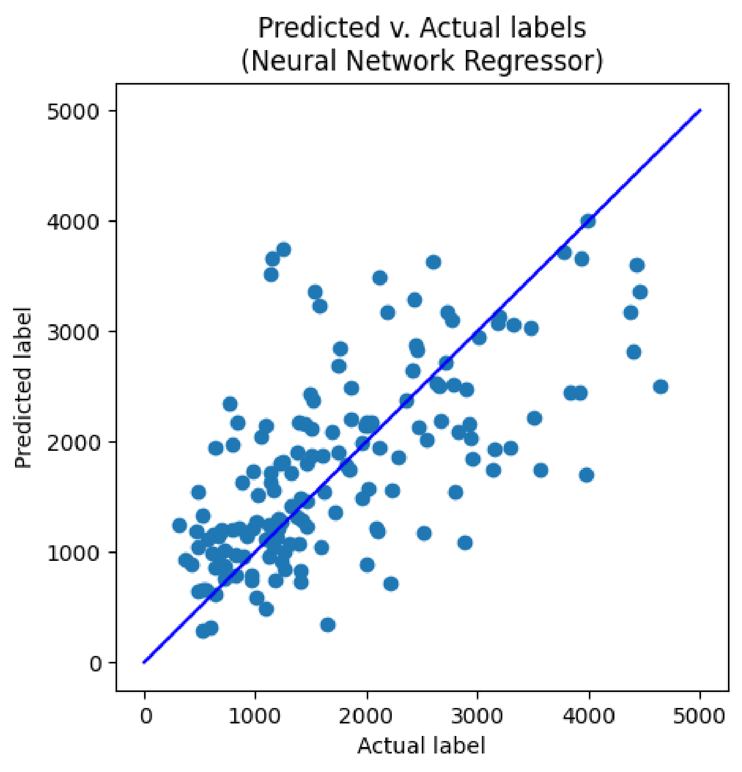
```
Epoch 45/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 479.4638 - mse: 431352.4688 - r2_score: 0.5675
Epoch 46/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 500.2470 - mse: 441527.3750 - r2_score: 0.5678
Epoch 47/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 448.0999 - mse: 391516.3750 - r2_score: 0.5813
Epoch 48/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 458.6790 - mse: 372378.8750 - r2_score: 0.6256
Epoch 49/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 467.5250 - mse: 401705.8438 - r2_score: 0.6099
Epoch 50/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 477.9524 - mse: 453176.8750 - r2_score: 0.5740
```

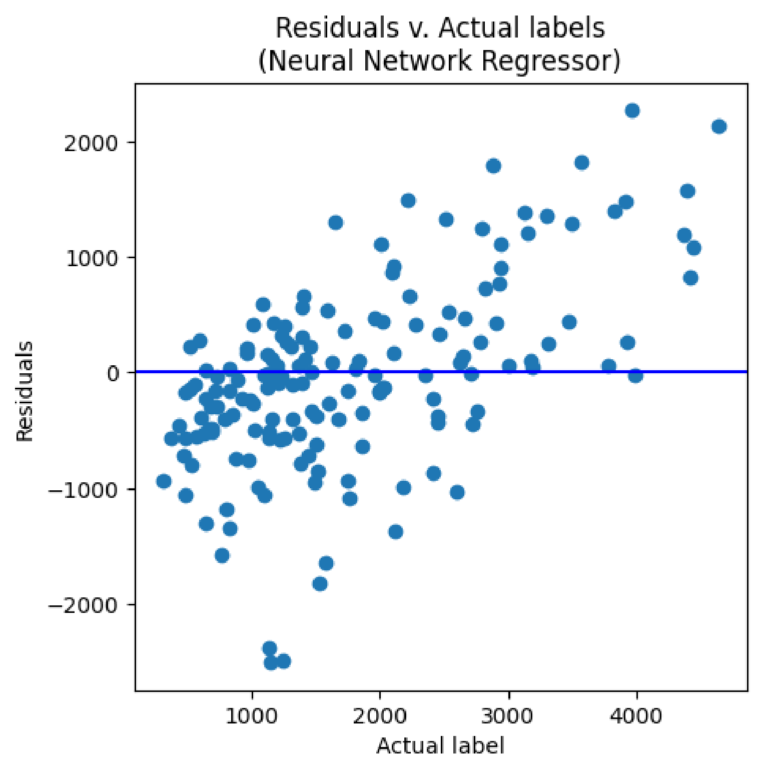
Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → **Model Training** → Evaluation → Deployment

Neural Network Regressor:





Analytics approach: Train a Regression Model

Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → **Evaluation** → Deployment

	Model	MSE	RMSE	MAE	r2
0	Classic Multiple Linear Regression	569932.198114	754.938539	577.504160	0.455478
1	Scaled Multiple Linear Regression	569932.198114	754.938539	577.504160	0.455478
2	Lasso Regression	550968.782255	742.272714	560.754248	0.473596
3	Decision Tree Regressor	854965.418559	924.643401	694.706381	0.183154
4	Random Forest Regressor	460978.376255	678.953884	494.255840	0.559575
5	Gradient Boosting Regressor	461953.866345	679.671881	492.132354	0.558643
6	Neural Network Regressor	661569.189551	813.369037	597.070978	0.367927

Considering the scoring metrics captured, the best model would be the Random Forest Regressor.

Analytics approach: Train a Regression Model

S&P Global
Commodity Insights

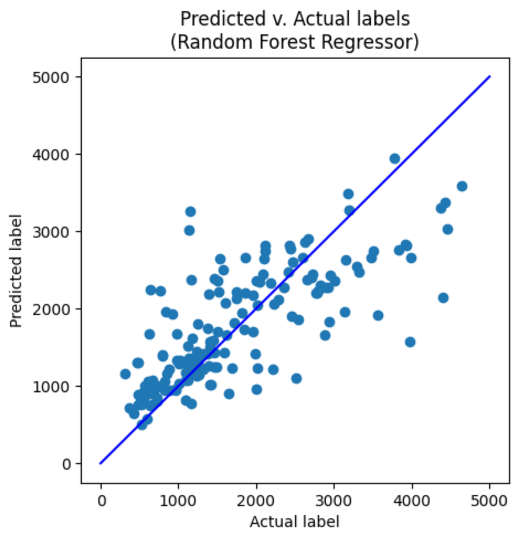
Data Collection → Data Cleaning → Exploratory Data Analysis (EDA) → Data Preparation → Model Training → Evaluation → **Deployment**

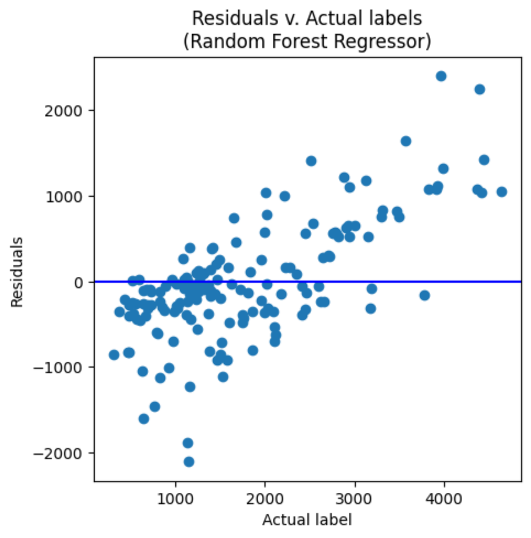
Random Forest Regressor:

3.8.10. Export best model for deployment and further optmization

```
import joblib

# Save the model as a pickle file
filename = './production_prediction_RF.pkl'
joblib.dump(model_rf, filename)
```

```
['./production_prediction_RF.pkl']
```





- Further hyperparameter tuning and cross validation is needed to optimize the model output.
- Also, there is a structure in the residual plot of all tested models. This is an indication of a polynomial relation between at least one feature and the target variable. Further exploration is needed to identify the feature that would provide the desired result.

S&P Global
Commodity Insights

Thank You!

Any Questions?

Observations and comments:

Further hyperparameter tuning and cross validation is needed to optimize the model output.

Also, there is a structure in the residual plot of all tested models. This is an indication of a polynomial relation between at least one feature and the target variable. Further exploration is needed to identify the feature that would provide the desired result.