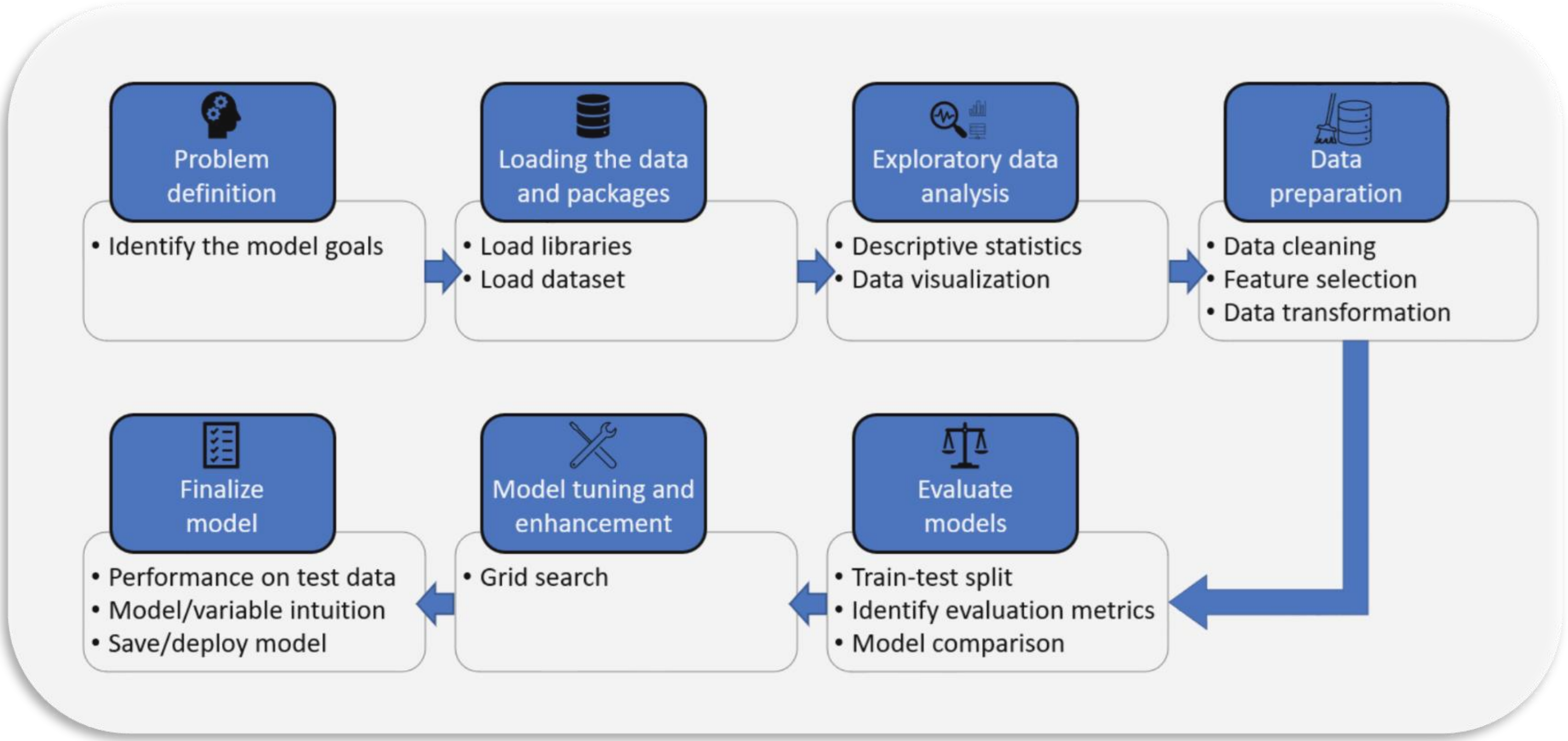
Petro.ai

Subsurface Assessment

CONTENT

- Problem Definition
- Loading data and packages
- Exploratory data analysis
- Data preparation
- Finalize model
- Model tuning and enhancements
- Evaluation models



PROBLEM DEFINITION

Problem definition

- Identify the model goals

- The goal of the machine learning model is to predict the wells cumulative 12-month oil production.

- The model is a function that predicts y given x_1, x_2,...x_i.

  - $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_i x_i$

- The target variable (y): is the cumulative 12-month oil produced.

- The input variables are: ['drainage_area', 'totalProppantByPerfLength', 'avgHzDistAnyZone', 'latitude', 'longitude', 'angleFromSHMax', 'lateralLength’]

Loading the Data and Packages

Loading the data and packages

- Load libraries
- Load dataset

- The libraries used to process and evaluate the dataset are:

  Pandas – Library for data manipulation. It offers data structures to handle tables and provides tools to manipulate them.

  NumPy – NumPy provides support for large, multidimensional arrays as well as a large collection of mathematical functions.

  Seaborn – A library for data visualization that is based on Matplotlib. It proves a high-level of interface for drawing attractive statistical graphics.

  Matplotlib – Matplotlib is a plotting library for creating 2D charts and lots.

  Sklearn – Sklearn is a library offering a wide range of machine learning algorithms and utilities.

- The csv file labeled 'well_stats_Delaware' is loaded into the pandas dataframe.

```
df = pd.read_csv('../data/part2/well_stats.csv')
```

- The first few rows of the data are viewed.
  - Df.head() – view the first 5 rows
  - column_list = df.columns.values.tolist() – list all columns in dataframe

Exploratory Data Analysis

Exploratory data analysis

- Descriptive statistics
- Data visualization

- General information about the dataset is gathered and repairs are made.
- The shape of the pandas dataframe is viewed:

  print(f"There are {df.shape[0]} rows and {df.shape[1]} columns in the raw well stats file.")

  There are 2457 rows and 172 columns in the raw well stats file.

  print(df_model.dtypes)

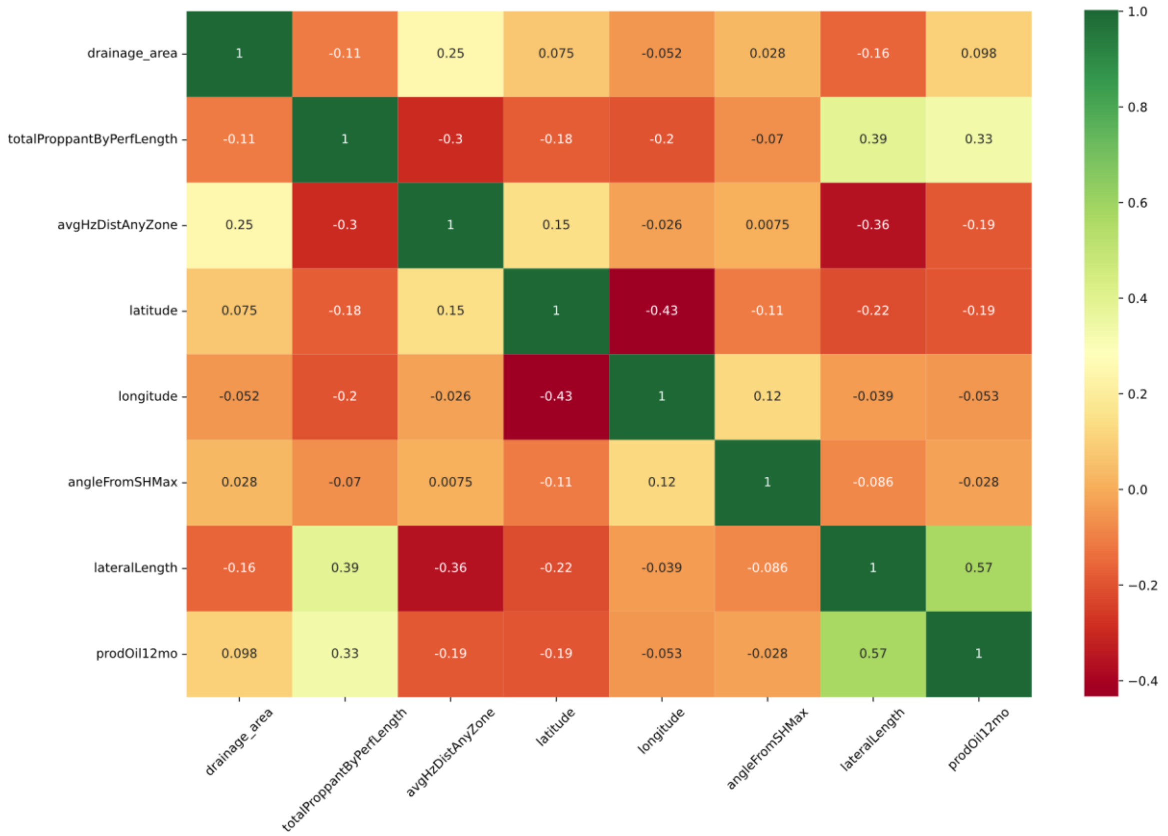
Column	dtype
drainage_area	Float64
totalProppant ByPerfLength	Float64
avgHzDistAnyZone	Float64
Latitude	Float64
Longitude	Float64
angleFromSHMax	int64
LateralLength	float64
prodOil12mo	float64

- The data is filtered to variables used for modeling and columns are renamed.

  df_model = df[['drainage_area', 'totalProppantByPerfLength', 'avgHzDistAnyZone', 'latitude', 'longitude', 'angleFromSHMax', 'lateralLength', 'prodOil12mo']]

- Plot correlation matrix as heatmap

Heatmap



Data Preparation

- Missing data is identified and removed.

```
df_model.isna().sum()
df_model = df_model[df_model['drainage_area'] > 0].copy()
df_model = df_model.dropna(inplace=False)
```

- Final dataframe included 1798 rows and 8 columns.

	drainage_area	totalProppantByPerfLength	avgHzDistAnyZone	latitude	longitude	angleFromSHMax	lateralLength	prodOil12mo
0	194838.58	2054.4040	1230.63430	36.389086	-113.386406	119	9247.0	132413.00
2	346702.40	1445.6741	2500.00000	36.371887	-113.457380	118	9163.0	185974.00
5	346830.94	458.1104	2500.00000	36.660776	-113.510260	126	4048.0	99082.00
6	342634.16	2830.5420	1471.00260	36.458529	-113.552025	110	8285.0	228579.00

- Data was Standardized and Normalized
- The target feature and variables were separated and scaled with sklearn's MinMaxScaler.

    The feature range was between 0 and 1.

- Lasso and Ridge Regression were normalized.
- Elastic Net was not changed.

Data preparation

- Data cleaning
- Feature selection
- Data transformation

Other scalers are available:
RobustScaler() if you have outliers, this scaler will reduce the effect the influece of outliers.
StandardScaler() for relatively Normal Distribution.
Normalizer() works on the rows, not the columns.

EVALUATE MODELS

Evaluate models

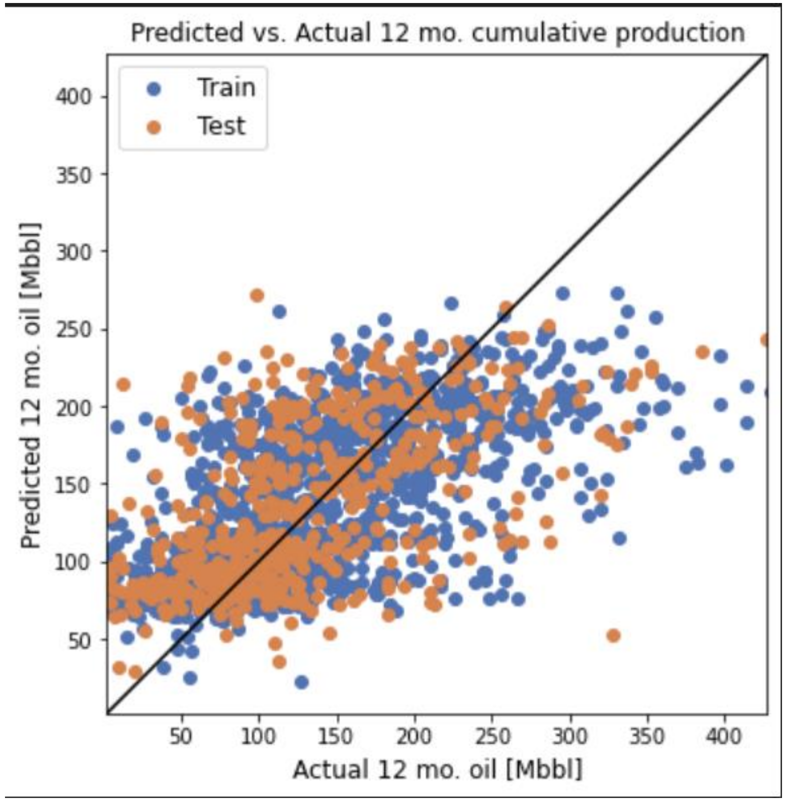
- Train-test split
- Identify evaluation metrics
- Model comparison

- All models were trained, tested, and split with sklearn.
  - Training Size: 70%
  - Test Size: 30%

Variables	Linear	Lasso	Ridge	Elastic
Y-intercept	-0.088	-1987904.000	-884887.447	-163062.558
drainage_area	0.291	0.324	0.129	0.327
totalProppantByPerfLength	0.423	11.552	9.606	13.301
avgHzDistAnyZone	0.018	3.354	-4.004	3.668
latitude	-0.037	-32973.070	-30427.075	-689.219
longitude	-0.042	-27077.930	-17851.484	-549.854
angleFromSHMax	0.040	120.572	24.822	128.041
lateralLength	0.453	17.647	8.308	17.889

R_Square	Linear	Lasso (alpha=0)	Ridge (alpha =1)	Elastic
Test_r2	0.305	.305	0.278	0.299
Training_r2	0.407	0.407	0.316	0.404

Linear Regression

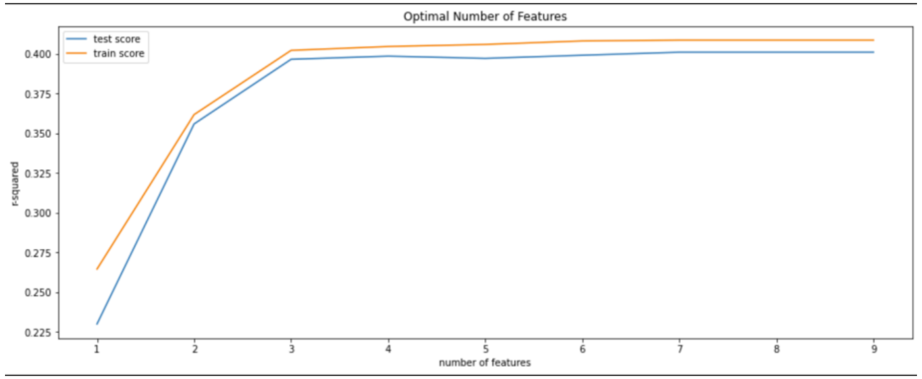


Model Tuning and Enhancement

Model tuning and enhancement

- Grid search

- Tuned number of parameters for linear regression
  - folds = KFold(n_splits = 5, shuffle = True, random_state = 100)
  - hyper_params = [{'n_features_to_select': list(range(1, 10))}]
  - specify model
  - lr2 = LinearRegression()
  - lr2.fit(X_train, y_train)
  - rfe = RFE(lr2)
  - model_cv = GridSearchCV(estimator = rfe, param_grid = hyper_params,
  - scoring= 'r2', cv = folds, verbose = 1,return_train_score=True)
  - model_cv.fit(X_train, y_train)



- Tuned alpha for Lasso and Ridge Model
  - Both Lasso and Ridge converge to Linear Model r2

R_Square	Linear	Lasso (alpha=0.001)	Ridge (alpha =0.001)	Elastic
Test_r2	0.305	.305	.305	0.299
Training_r2	0.407	0.407	0.407	0.404

params = {'alpha': [0.0001, 0.001, 0.01, 0.05, 1.0, 5.0, 10, 50, 100]}

ridge = Ridge()

folds = 5

grid_cv_model = GridSearchCV(estimator=ridge, param_grid=params, scoring='neg_mean_absolute_error', cv=folds, return_train_score=True, verbose=1)

grid_cv_model.fit(X_train,y_train)

grid_cv_model.best_params_

Best Parameter {'alpha': 0.0001}

Finalize MODEL

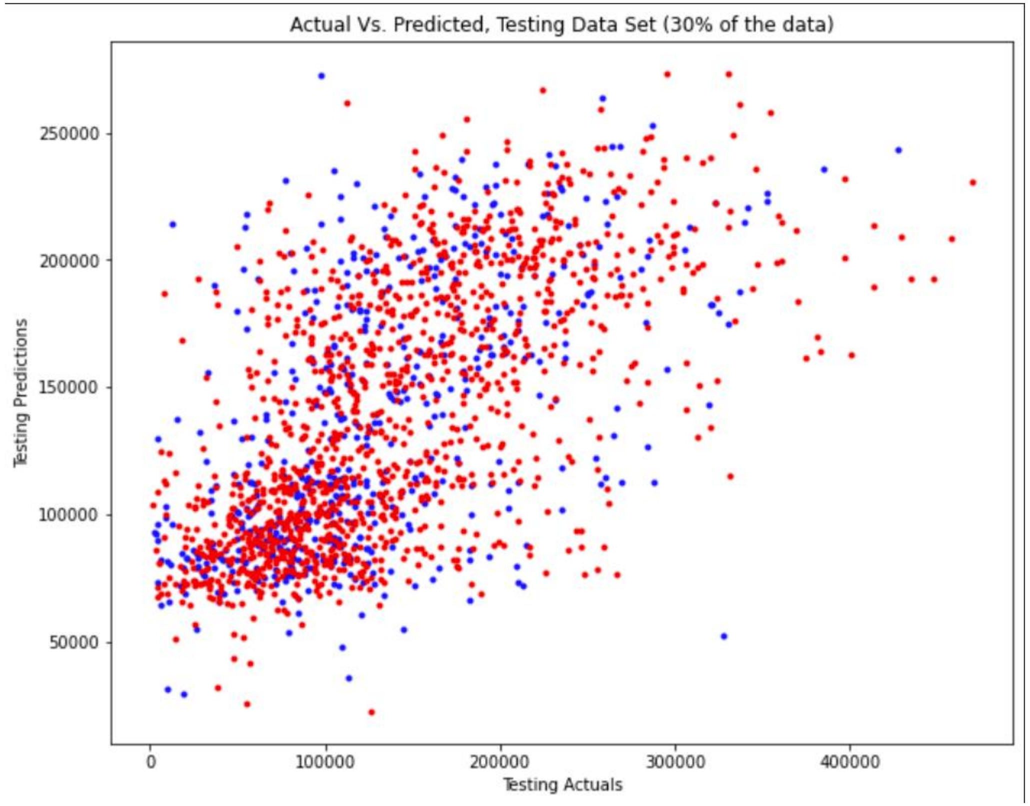
Finalize model

- Performance on test data
- Model/variable intuition
- Save/deploy model

- Linear model provides the best fit.

Variables	Linear	Lasso	Ridge	Elastic
Y-intercept	-0.088	-1987904.000	-884887.447	-163062.558
drainage_area	0.291	0.324	0.129	0.327
totalProppantByPerfLength	0.423	11.552	9.606	13.301
avgHzDistAnyZone	0.018	3.354	-4.004	3.668
latitude	-0.037	-32973.070	-30427.075	-689.219
longitude	-0.042	-27077.930	-17851.484	-549.854
angleFromSHMax	0.040	120.572	24.822	128.041
lateralLength	0.453	17.647	8.308	17.889

R_Square	Linear	Lasso (alpha=0)	Ridge (alpha =1)	Elastic
Test_r2	0.305	.305	0.278	0.299
Training_r2	0.407	0.407	0.316	0.404



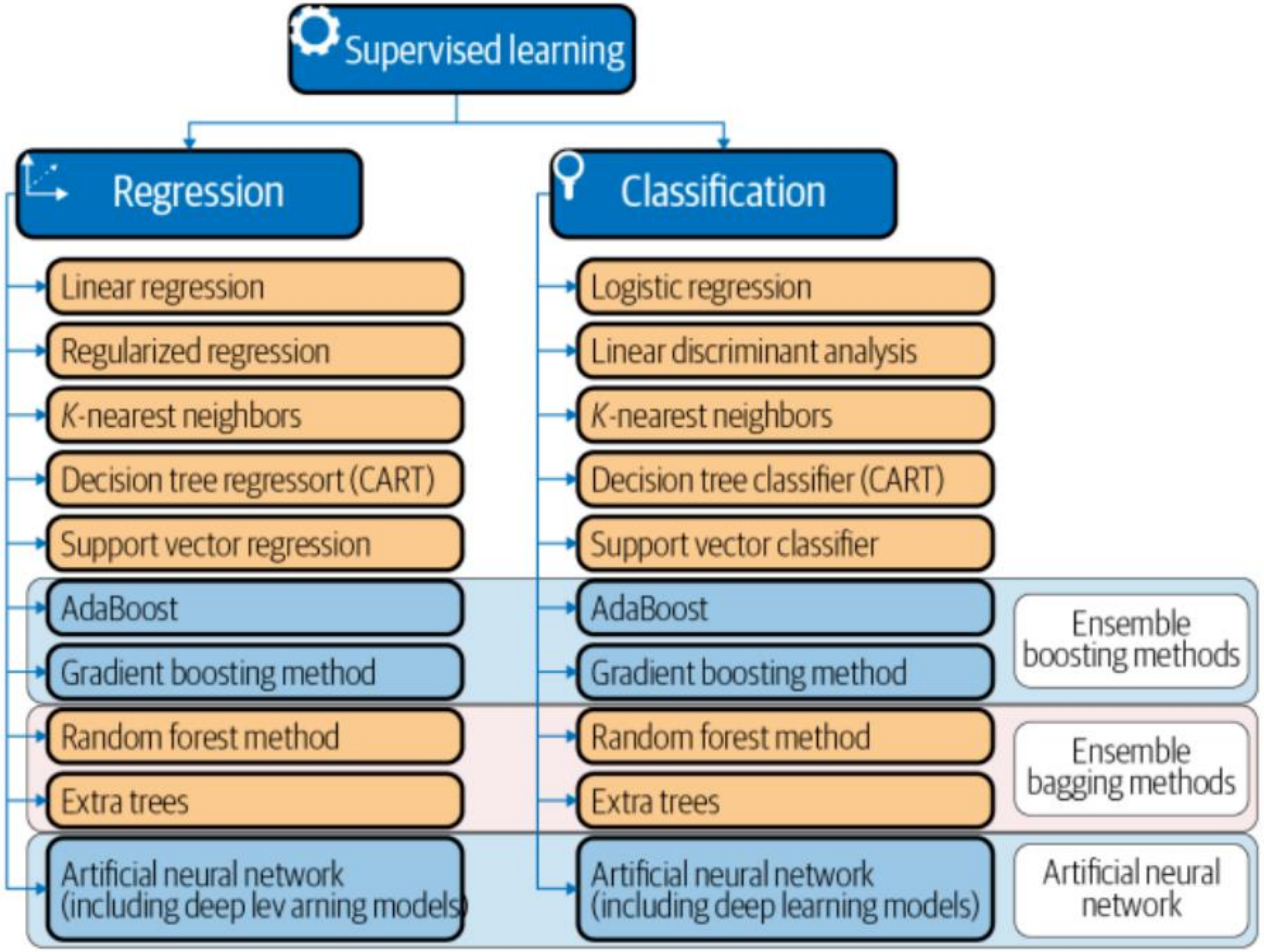
END

Next Steps

- Deep dive into data

- SGD (Stochastic Gradient Descent)
- K-Nearest Neighbors (KNN)
- Time series (LSTM)
- Spatial relationships

	Linear regression	Logistic regression	SVM	CART	Gradient boosting	Random forest	Artificial neural network	KNN	LDA
Simplicity	✓	✓	✓	✓	✗	✗	✗	✓	✓
Training Time	✓	✓	✗	✓	✗	✗	✗	✓	✓
Handle non-linearity	✗	✗	✓	✓	✓	✓	✓	✓	✓
Robust to overfitting	✗	✗	✓	✗	✗	✓	✗	✓	✗
Large datasets	✗	✗	✗	✓	✓	✓	✓	✗	✓
Many features	✗	✗	✓	✓	✓	✓	✓	✗	✓
Model interpretation	✓	✓	✗	✓	✓	✓	✗	✓	✓
Feature scaling needed	✗	✗	✓	✗	✗	✗	✗	✗	✗

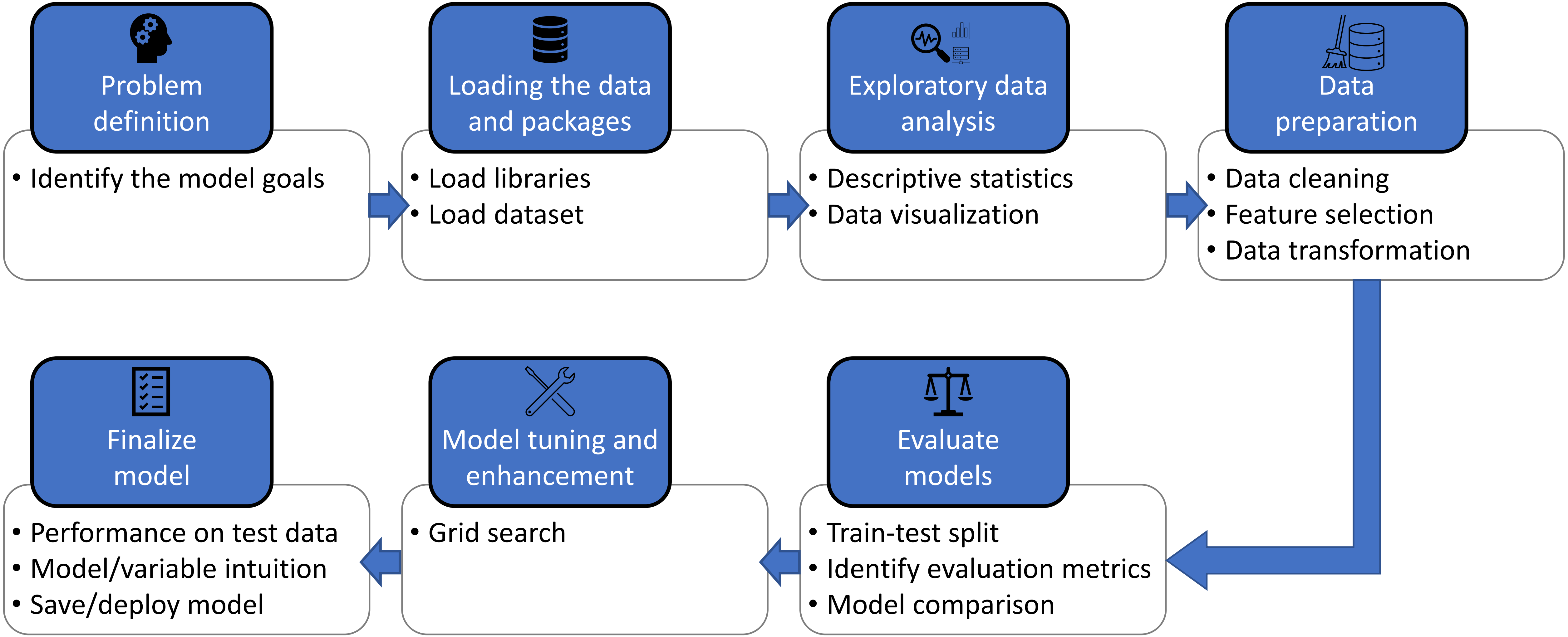


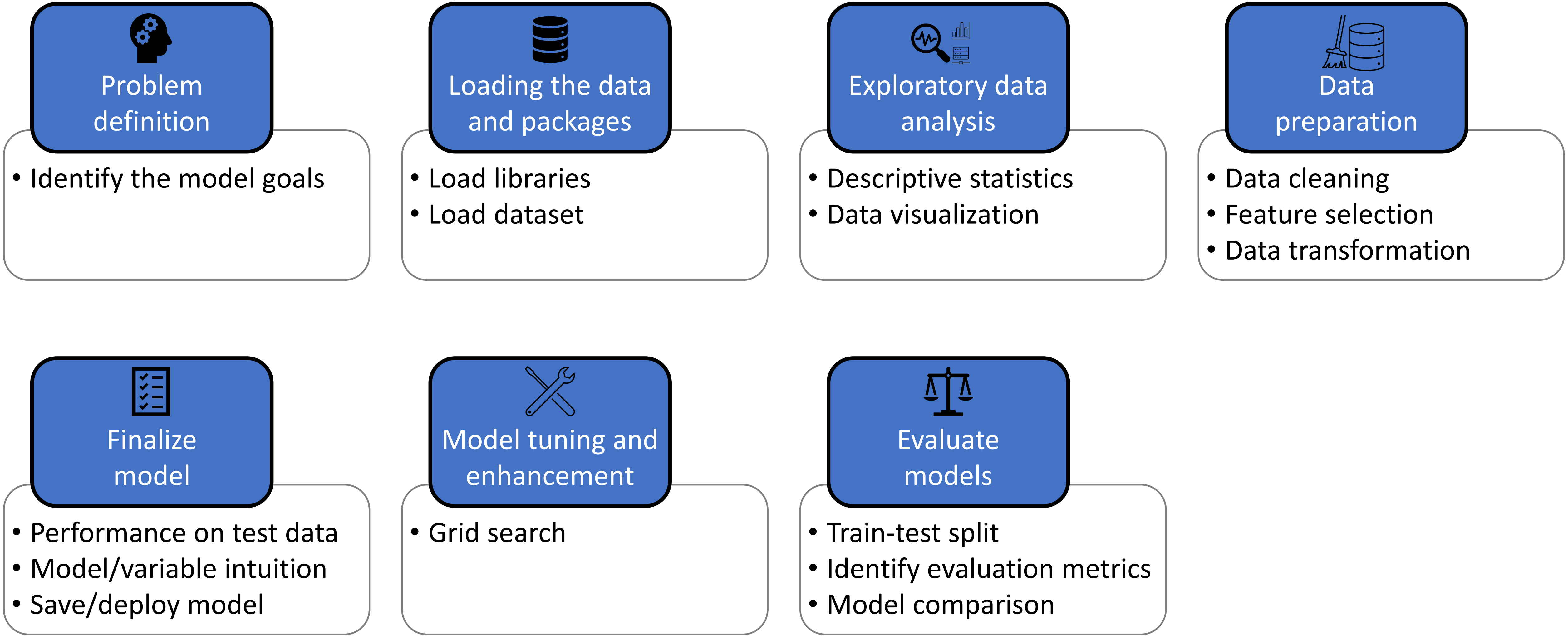
Regression

- Mean absolute error (MAE)
- Mean square error (MSE)
- R squared (R^2)
- Adjusted R squared (Adj- R^2)

Classification

- Accuracy
- Precision
- Recall
- Area under curve (AUC)
- Confusion matrix


Problem definition
• Identify the model goals
Loading the data and packages
• Load libraries
• Load dataset
Exploratory data analysis
• Descriptive statistics
• Data visualization
Data preparation
• Data cleaning
• Feature selection
• Data transformation
Evaluate models
• Train-test split
• Identify evaluation metrics
• Model comparison
Model tuning and enhancement
• Grid search
Finalize model
• Performance on test data
• Model/variable intuition
• Save/deploy model


Problem definition
• Identify the model goals
Loading the data and packages
• Load libraries
• Load dataset
Exploratory data analysis
• Descriptive statistics
• Data visualization
Data preparation
• Data cleaning
• Feature selection
• Data transformation
Finalize model
• Performance on test data
• Model/variable intuition
• Save/deploy model
Model tuning and enhancement
• Grid search
Evaluate models
• Train-test split
• Identify evaluation metrics
• Model comparison

	Drainage_area	totalProppant ByPerfLength	avgHzDistAnyZone	latitude	longitude	angleFrom SHMax	Lateral Length	prodOil12mo
Drainage_area	1.000000	-0.079115	0.204507	0.05661	-0.048259	0.020854	-0.091635	0.108534
totalProppant ByPerfLength	-0.079115	1.000000	-0.297495	-0.16446	-0.199341	-0.061656	0.385214	0.335033
avgHzDistAnyZone	0.204507	-0.297495	1.000000	0.15475	-0.014669	-0.010518	-0.383163	-0.218147
latitude	0.056612	-0.164460	0.154756	1.00000	-0.450329	-0.105174	-0.218155	-0.185327
longitude	-0.048259	-0.199341	-0.014669	-0.45032	1.000000	0.110571	-0.036311	-0.045839
angleFromSHMax	0.020854	-0.061656	-0.010518	-0.10517	0.110571	1.000000	-0.051407	-0.031307
LateralLength	-0.091635	0.385214	-0.383163	-0.21815	-0.036311	-0.051407	1.000000	0.588168
prodOil12mo	0.108534	0.335033	-0.218147	-0.18532	-0.045839	-0.031307	0.588168	1.000000