

PROJECT I

PROBLEM STATEMENT:

“Prediction of facies data from the provided well log data using Machine Learning algorithms”

Objective:

From the given well log data, we build and train a model considering the required features and finally deploy it. This model should serve the main purpose of predicting the facies value with considerable accuracy. This is a classification problem that can be solved by implementing the required ML algorithms.

Data Collection:

The first step in the process of building a model after defining the problem is to collect the data. Data collection is a crucial step in any data-driven project, including machine learning. It involves gathering relevant data that will be used to train, validate, and test your ML model. Depending on the data sources, you may use various methods for data collection, including web scrapping, data entry, sensor data and data mining.

In this project, the data has been gathered from sensors, IoT devices, or other data-capturing technologies. This data is called sensor data. This data is collected dynamically from the oil wells, while the drilling process is going on which is why it is also called well log data. This data is originally in the binary format or as las files with ‘.las’ extension. For building our model, we need this data in an accessible format which is generally the spreadsheet format with a ‘.csv’ format. We use a plugin which changes the las files into csv files which makes them accessible.

Then, the next step would be data pre-processing.

Data pre-processing:

Data preprocessing is a crucial step in preparing the collected data for machine learning tasks. It involves cleaning, transforming, and organizing the data to improve its quality and compatibility with the chosen ML algorithms. Initially let us understand the important features of the well log data.

Key features:

Facies: In well data, the term "facies" refers to distinct rock or sedimentary units with similar characteristics that were deposited in a specific environment or geological setting. Facies analysis is an important aspect of subsurface geology, particularly in petroleum exploration and production, as it provides valuable information about reservoir properties, depositional environments, and stratigraphic relationships. This is our target variable.

Gamma: In well data, "gamma" typically refers to the gamma ray measurement obtained from well logging tools. Gamma ray logging is a common technique used in the oil and gas industry to gather information about the geological formations penetrated by a wellbore.

VSH: In well data, "Vsh" stands for Volume of Shale. Vsh is a parameter used to quantify the proportion or percentage of shale within a formation or interval of interest along a wellbore. It is an important parameter in petrophysical analysis and reservoir characterization.

Porosity: In well data, "porosity" refers to the percentage of void spaces or pores within a rock or sedimentary formation. It is a fundamental parameter used to evaluate the ability of a reservoir rock to store and transmit fluids such as oil, gas, or water. Porosity is a critical property in reservoir characterization and plays a significant role in assessing the quality and productivity of hydrocarbon reservoirs.

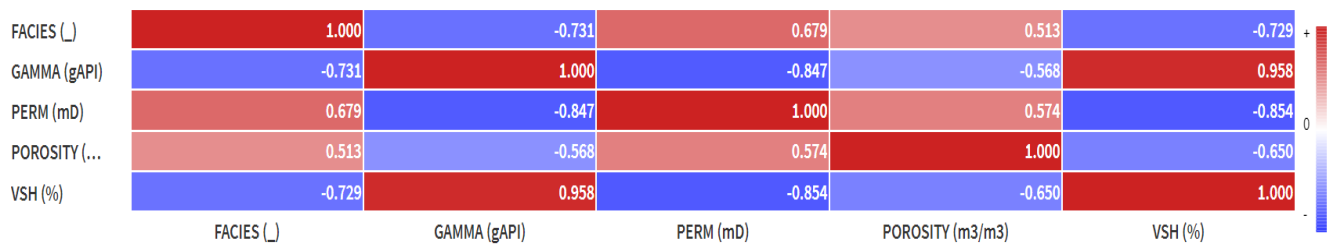
Permeability: Permeability is a crucial parameter in well data analysis as it helps determine the ability of a reservoir rock to transmit fluids, such as oil, gas, or water. Permeability is typically measured in units of darcies (D) or millidarcies (mD).

Well: This would be name of the well which could be in the range of C1-C6. This is subjective to the organization handling the wells.

These features can be considered significant in predicting the target variable. We should note that there are other features that are not explained here which do not play a vital role in determining the result. They can be handled in this step. So, the first step in data pre-processing is to remove the features which do not play a significant role in the model building due to high correlation with the target variable.

Let us look at the correlation matrix of the selected features.

Correlation matrix on 5 variables (Spearman) No split



The removed columns are caliper, depth, net gross, net perm, true vertical thickness etc. This step is done by using the prepare recipe in the dataiku platform and removing all the columns one by one. Then, all the insignificant columns are removed from the dataset. Then we add a new step in the prepare recipe and use the “Remove rows where cells are empty in the column:” option. We now input each significant feature columns separately to eliminate all the empty cells in the dataset. This entire process is called **data cleaning**.

The next step here is to remove the outliers. We click on the analyze tab for all the continuous features and remove the outliers which are out of 5 iqr range. This concludes the data pre-processing.

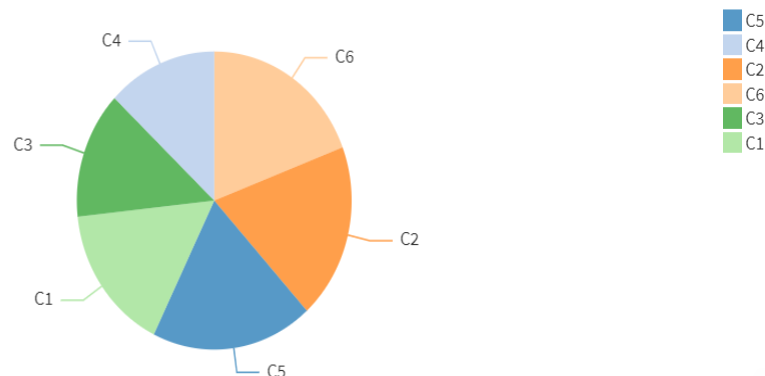
Data splitting:

Data splitting is a crucial step in machine learning to evaluate and validate the performance of a trained model. The dataset is divided into separate subsets to train the model, tune hyperparameters, and evaluate its generalization on unseen data. We use the train-test split method here. All the data belonging to the well “C6” will be considered as test data and the data of the wells from “C1” to “C5” form the train set

Avg of FACIES (L) by Well

33480 records

Run on DSS



As we can see, the above pie chart shows the distribution of our target variable across different wells. From here we use the split recipe to split the data set into train and test sets. In the recipe we select the option that says “Splitting based on the values of a single column”. There we give well as the selected feature and assign “C6” to the test set and the rest to train set. We will now get two datasets as a result of this recipe. The next step is to select a model and build it on the train set.

Model Selection:

Our goal is to predict the facies value using other input variables. So now, we go to the flow and in the Flow, select the pre-processed dataset, and click on the Lab button in the right panel. In Dataiku, a "lab" refers to a feature called Dataiku Lab, which is an interactive coding environment within the platform. With Dataiku Lab, you can perform various data-related tasks such as data exploration, data cleaning, feature engineering, model development, and deploying machine learning models.

We select the “Auto-ML prediction” option and then click create. This will create an auto prediction models with few already selected algorithms. You can alter these details in the design tab. After going to the design tab, we can do feature handling i.e. selecting the input features or create features from existing features.

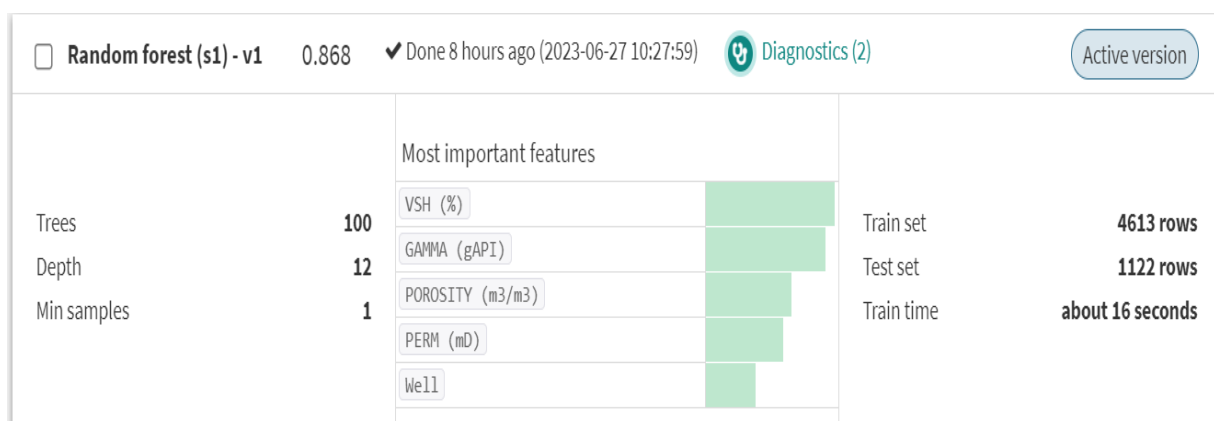
Then we can go to the algorithms section and select what all Machine Learning algorithms should be used or implemented while building our model. Let us look at what some important ML algorithms and how they work.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

Logistic regression is a popular Machine Learning algorithm, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that combines the principles of gradient boosting and gradient descent optimization. It forms an ensemble of weak learners, usually decision trees, to create a strong predictive model. By iteratively training these trees and minimizing a specific loss function, XGBoost efficiently handles complex data patterns and provides accurate predictions.


Then, dataiku will train the data into models using the selected algorithms and also provide their R2 score beside them. The R-squared (R2) score is a commonly used evaluation metric in machine learning for regression tasks. It measures the goodness of fit of a regression model and represents the proportion of the variance in the dependent variable that can be explained by the independent variables. The R2 score ranges from 0 to 1, with a higher value indicating a better fit of the model to the data. An R2 score of 1 indicates that the model perfectly predicts the dependent variable, while an R2 score of 0 suggests that the model does not provide any predictive power beyond simply using the mean of the dependent variable. We select the algorithm with higher R2 score value and build the model using that algorithm. In this case, Random Forest algorithm has the highest R2 score so we select the algorithm and train the model using Random Forest. As we can see the R2 score is 0.868 which implies that it is producing good results.



After we view the required options in the design tab and modify them, we train the model. This will take us to the next step which is model training.

Model Training:

Now, we select the Random Forest method and train the model in that algorithm.

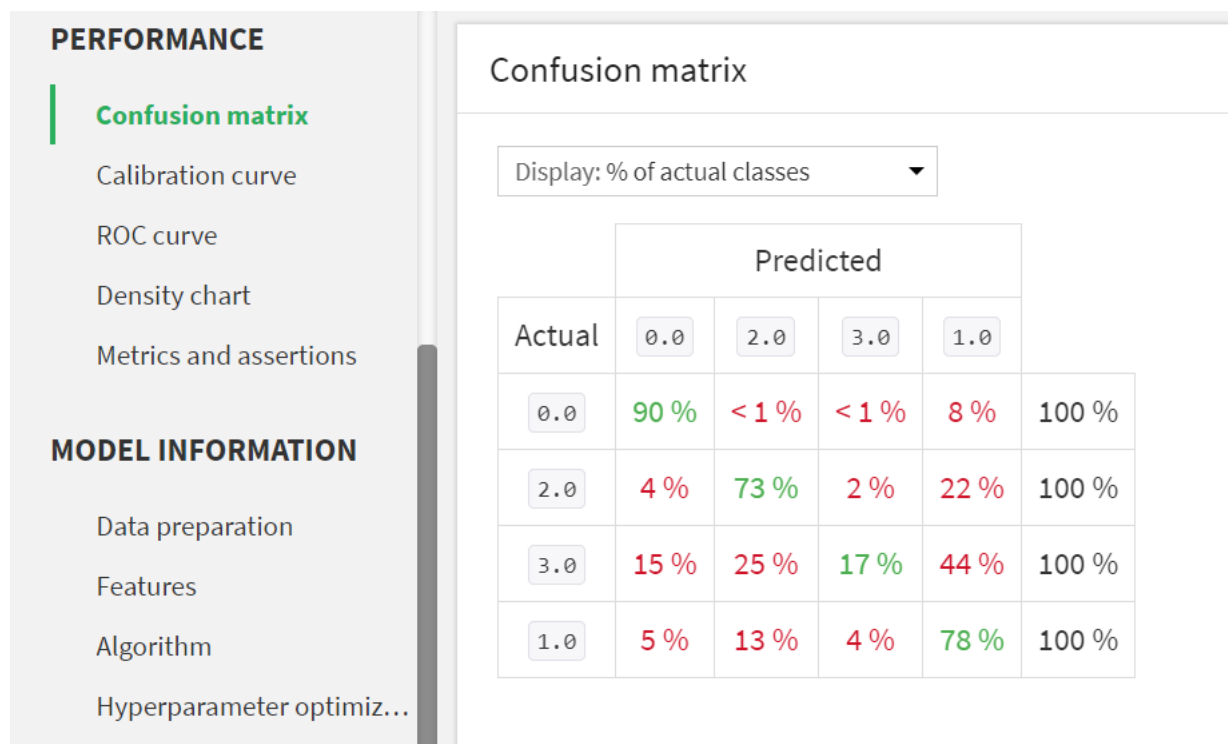
 Model	
Model ID	S-LAS_FILES-b4oZq4rW-initial
Model type	Multiclass classification
Target	FACIES (_)
Classes	0.0 2.0 3.0 1.0
Backend	Python (in memory)
Algorithm	Random forest classification
Trained on	2023/06/27 10:27
Columns	6
Train set rows	4613
Test set rows	1122

The model type is multi class classification and the target variable is facies. This image shows other information about the model.

We will now look at the model performance in the performance tab. Let us look at the confusion matrix of our model.

A confusion matrix is a performance evaluation tool used in machine learning classification tasks to visualize and analyze the performance of a model by comparing predicted and actual class labels. It provides a comprehensive summary of the model's predictions and helps assess its accuracy, precision, recall, and other metrics. The confusion matrix is typically represented in a tabular format.

The confusion matrix provides a detailed understanding of the model's performance, helping identify potential errors and imbalances in predictions. It is a valuable tool for evaluating and comparing different machine learning models and selecting the most suitable one based on the desired trade-offs between precision, recall, and other metrics.



In this project, we are more concerned about predicting '0' and in 90% of the cases 0 is being predicted as 0. So, this can be considered as a good model and be sent to evaluation. Model evaluation is the next step in the process.

Model Evaluation:

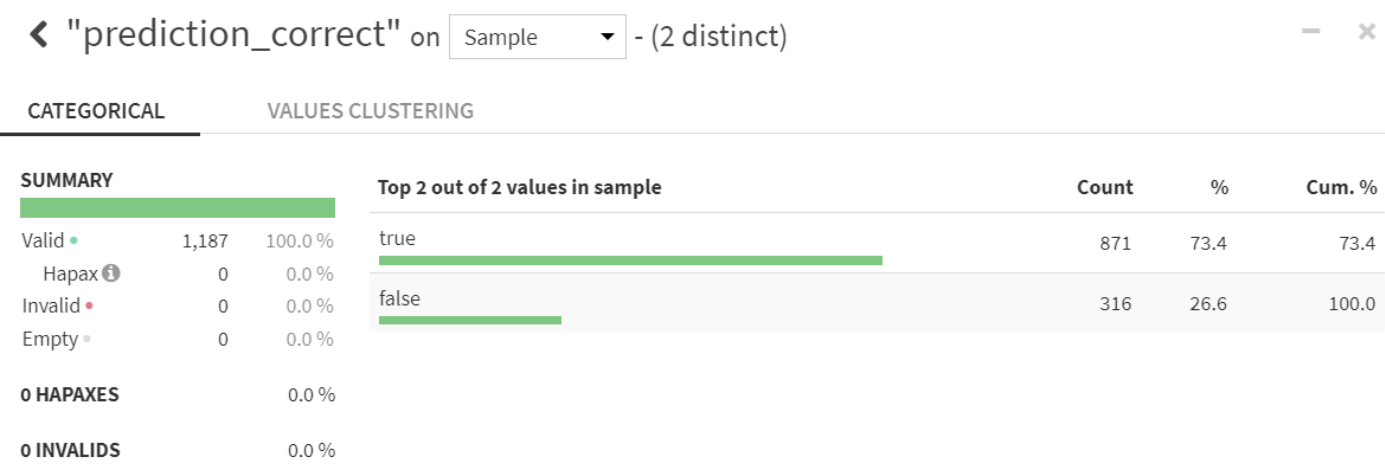
Model evaluation in machine learning involves assessing the performance and effectiveness of a trained model on unseen data. It helps determine how well the model generalizes to new instances and provides insights into its strengths, weaknesses, and areas for improvement. It is important to select evaluation metrics appropriate for the specific problem domain and the goals of the model.

We use the evaluate recipe provided by dataiku to perform model evaluation. The recipe needs to inputs, one being the input dataset and other being the model that must be evaluated. Here, the input dataset is the test data which contains well data of the well 'C6' and the model that we trained will be the input model. Then we run the evaluate recipe which gives us an output dataset with predicted data.

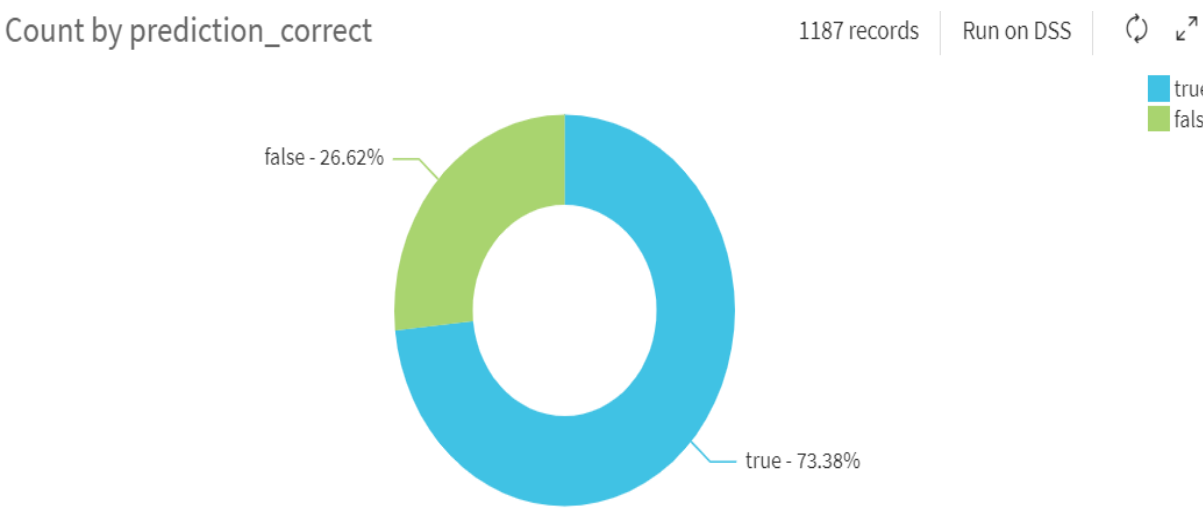
In this case, the output dataset will contain a column of predicted facies value and another column which shows whether the prediction is true or not. If there is a larger percentage of true

values in that column, that implies that the model we have created can be deployed and be used for further uses and vice versa.

A small peek into the analysis of the column will show us the model's performance and effectiveness, so let us have a look at it.



As we can see, the facies value is predicted correctly 73.4% of the time which tells us that it performs decently. This test data contains well data of a single well, which is not included in the train set. If the model is able to perform well and predict properly for this data, that implies that it would display a similar behavior with new datasets which need the help of this model.



This doughnut chart shows the accuracy of the model, which predicts correct data 73.38% of the times on this dataset.

Classification in machine learning involves categorizing input data into predefined classes. Our aim in this case is to see to it that we classify each data point into either of the four possible facies classes. Our model does this 73.8% of the time on this test data. This can be deployed.

Value addition:

Value addition is the value added to the domain by this model. This ML model adds good value to the domain. It largely helps in cost reduction. The collection of this data is done by equipment which include sensors, drill bits and other instruments which cost a large amount of money and labor for the process.

In a scenario where we have 100 wells for which facies must be calculated, it would be highly expensive and laborious to do it manually by using equipment. Instead, we could use our model here. We can find facies manually for 15-20 wells manually and collect the data properly with all other parameters. This data can be used to build a model which predicts facies using algorithms which classify new data points using their input/ independent variables and classify them into either of the facies class.

From the above scenario, it is visibly evident that this model helps largely and cost cutting and making the task and reduces the use of the equipment and labor for all the wells. In this manner, this model adds large value to the organization,

PROJECT II

Problem Definition:

Log Reconstruction-Predicting Missing Log Data Using Available data.

Objective:

From the given well log data, we build and train a model considering the required features and finally deploy it. This model should serve the main purpose of predicting the facies value with considerable accuracy. This is a Regression problem that can be solved by implementing the required ML algorithms.

Data Collection:

The first step in the process of building a model after defining the problem is to collect the data. Data collection is a crucial step in any data-driven project, including machine learning. It involves gathering relevant data that will be used to train, validate, and test your ML model. Depending on the data sources, you may use various methods for data collection, including web scraping, data entry, sensor data and data mining.

In this project, the data has been gathered from sensors, IoT devices, or other data-capturing technologies. This data is called sensor data. This data is collected dynamically from the oil wells, while the drilling process is going on which is why it is also called well log data. This data is originally in the binary format or as las files with '.las' extension. For building our model, we need this data in an accessible format which is generally the spreadsheet format with a '.csv' format. We use a plugin which changes the las files into csv files which makes them accessible. Then, the next step would be data pre-processing.

Data Preprocessing-:

Features:

1. DEPTH: In the context of well data, the term "depth" refers to the vertical measurement of a particular point or interval within a wellbore. It is a crucial parameter used in the oil and gas industry to describe and analyze subsurface formations, drilling operations, and reservoir characteristics.

2. FACIES: In well data, the term "facies" refers to distinct rock or sedimentary units with similar characteristics that were deposited in a specific environment or geological setting.

Facies analysis is an important aspect of subsurface geology, particularly in petroleum exploration and production, as it provides valuable information about reservoir properties, depositional environments, and stratigraphic relationships.

3. GAMMA (gAPI): In well data, "gamma" typically refers to the gamma ray measurement obtained from well logging tools. Gamma ray logging is a common technique used in the oil and gas industry to gather information about the geological formations penetrated by a wellbore.

4. POROSITY(m3/m3): In well data, "porosity" refers to the percentage of void spaces or pores within a rock or sedimentary formation. It is a fundamental parameter used to evaluate the ability of a reservoir rock to store and transmit fluids such as oil, gas, or water. Porosity is a critical property in reservoir characterization and plays a significant role in assessing the quality and productivity of hydrocarbon reservoirs.

5. PERM (mD): In well data analysis, permeability refers to the ability of a rock or reservoir to transmit fluids, such as oil, gas, or water, through its interconnected pore spaces. It is a measure of the ease with which fluids can flow through the rock.

6. VSH (%): In well data, "Vsh" stands for Volume of Shale. Vsh is a parameter used to quantify the proportion or percentage of shale within a formation or interval of interest along a wellbore. It is an important parameter in petrophysical analysis and reservoir characterization.

7. WELL: This would be the name of the well which could be in the range of C1-C6. This is subjective to the organization handling the wells.

These features can be considered significant in predicting the target variable. We should note that there are

other features that are not explained here which do not play a vital role in determining the result. They can

be handled in this step. So, the first step in data pre-processing is to remove the features which do not play a

significant role in the model building due to high correlation with the target variable. The removed columns are caliper, net gross, net perm, true vertical thickness etc. This step is done by using the prepare recipe in the dataiku platform and removing all the columns one by one. Then, all the insignificant columns are removed from the dataset. Then we add a new step in the prepare recipe and use the "Remove rows where cells are empty in the column: " option. We now input each significant feature columns separately to eliminate all the empty cells in the dataset. This entire process is called data cleaning.

Data Splitting:

Data splitting is a crucial step in machine learning to evaluate and validate the performance of a trained model. The dataset is divided into separate subsets to train the model, tune hyperparameters, and evaluate its generalization on unseen data. We use the train-test split method here. All the data belonging to the well “C6” will be considered as test data and the data of the wells from “C1” to “C5” form the train set

From here we use the split recipe to split the data set into train and test sets. In the recipe we select the option that says “Splitting based on the values of a single column”. There we give well as the selected feature and assign “C6” to the test set and the rest to train set. We will now get two datasets as a result of this recipe. The next step is to select a model and build it on the train set.

Model Selection:

Lab in Dataiku:

In Dataiku, a "lab" refers to a feature called Dataiku Lab, which is an interactive coding environment within the platform. With Dataiku Lab, you can perform various data-related tasks such as data exploration, data cleaning, feature engineering, model development, and deploying machine learning models. It provides access to a wide range of libraries and tools commonly used in data science, including pandas, scikit-learn, TensorFlow, PyTorch, and more.

Our goal is to predict the Gamma value using other input variables. So now, we go to the flow and in the Flow, select the pre-processed dataset, and click on the Lab button in the right panel. In Dataiku, a "lab" refers to a feature called Dataiku Lab, which is an interactive coding environment within the platform. With Dataiku Lab, you can perform various data-related tasks such as data exploration, data cleaning, feature engineering, model development, and deploying machine learning models. We select the “Auto-ML prediction” option and then click create. This will create an auto prediction models with few already selected algorithms. You can alter these details in the design tab. After going to the design lab, we can do feature handling i.e. selecting the input features or create features from existing features.

Then we can go to the algorithms section and select what all Machine Learning algorithms should be used or implemented while building our model.

Let us look at some important ML algorithms and how they work:

1.Random Forest: Random Forest is a popular machine learning algorithm that belongs to the supervised

learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

2.Ridge Regression(L2): Ridge regression, also known as Tikhonov regularization or L2 regularization, is a linear regression technique that adds a penalty term to the ordinary least squares (OLS) objective function. This penalty term helps address the issue of multicollinearity and can improve the performance of the regression model. In ordinary least squares regression, the goal is to minimize the sum of squared residuals between the predicted values and the actual values. However, when the dataset contains highly correlated features, OLS can become sensitive to small changes in the input data, leading to unstable and overfit models.

Then, dataiku will train the data into models using the selected algorithms and also provide their R2 score beside them. The R-squared (R2) score is a commonly used evaluation metric in machine learning for regression tasks. It measures the goodness of fit of a regression model and represents the proportion of the variance in the dependent variable that can be explained by the independent variables. The R2 score ranges from 0 to 1, with a higher value indicating a better fit of the model to the data. An R2 score of 1 indicates that the model perfectly predicts the dependent variable, while an R2 score of 0 suggests that the model does not provide any predictive power beyond simply using the mean of the dependent variable. We select the algorithm with higher R2 score value and build the model using that algorithm. In this case, Random Forest algorithm has the highest R2 score so we select the algorithm and train the model using Random Forest. As we can see the R2 score is 0.885 which implies that it is producing good results.

Random forest (Model)		0.889	Done 18 hours ago (2023-06-27 15:53:22)	Diagnostics (1)	
Trees	100	Most important features			
Depth	12	PERM (mD)	<div></div>	Train set	4617 rows
Min samples	7	FACIES ()	<div></div>	Test set	1123 rows
Hyperparameter search size	2	POROSITY (m3/m3)	<div></div>	Train time	about 8 seconds


After we view the required options in the design tab and modify them, we train the model. This will take us to the next step which is model training.

Model Training and Deployment:

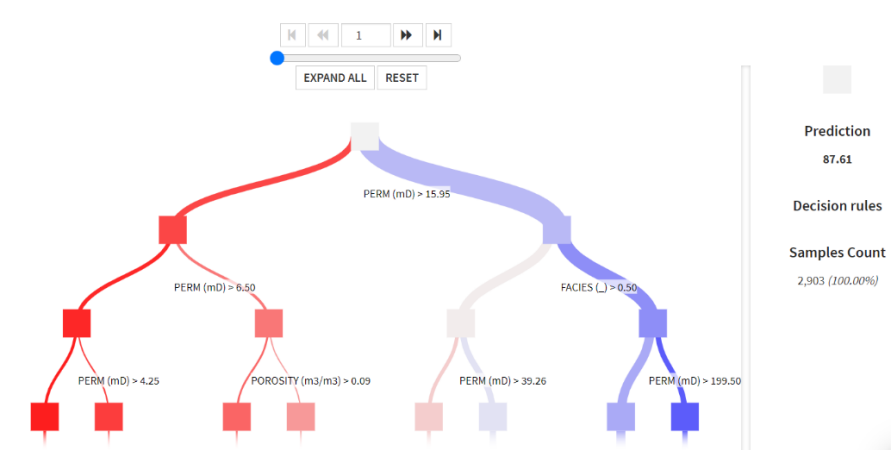
Random Forest:

Random forest (Model) 

R2 Score: 0.889

 Model	
Model ID	A-R1-aAJhhTVs-vnsdy2Ad-s1-pp1-m1
Model type	Regression
Target	GAMMA (gAPI)
Backend	Python (in memory)
Algorithm	Random forest regression
Trained on	2023/06/27 15:53
Columns	8
Train set rows	4617
Test set rows	1123
Calibration method	No calibration
Code Env	DSS builtin env

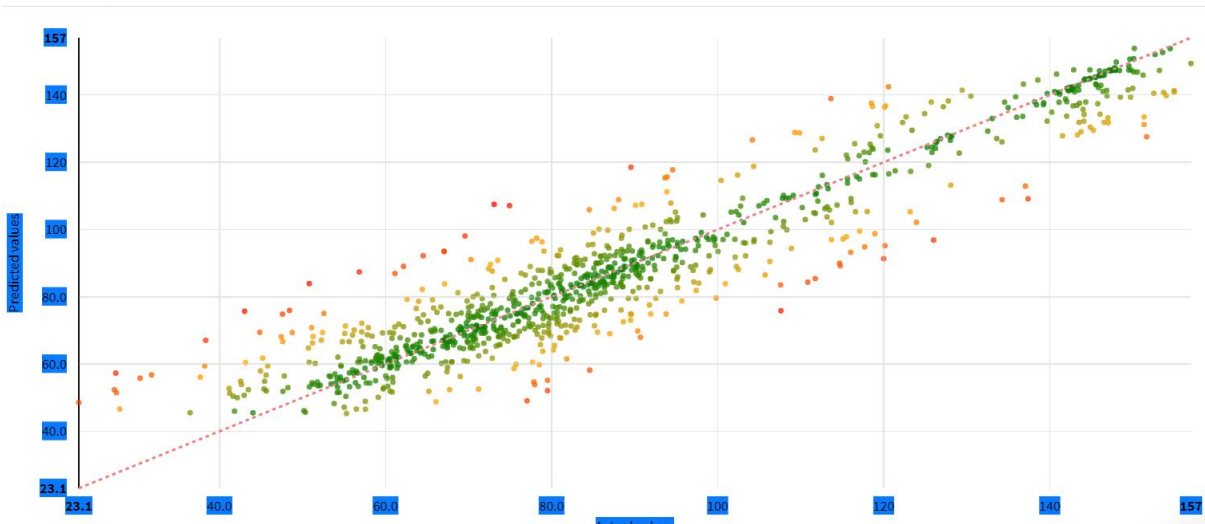
Decision Trees:



Performance:

1.Scatter Plot:

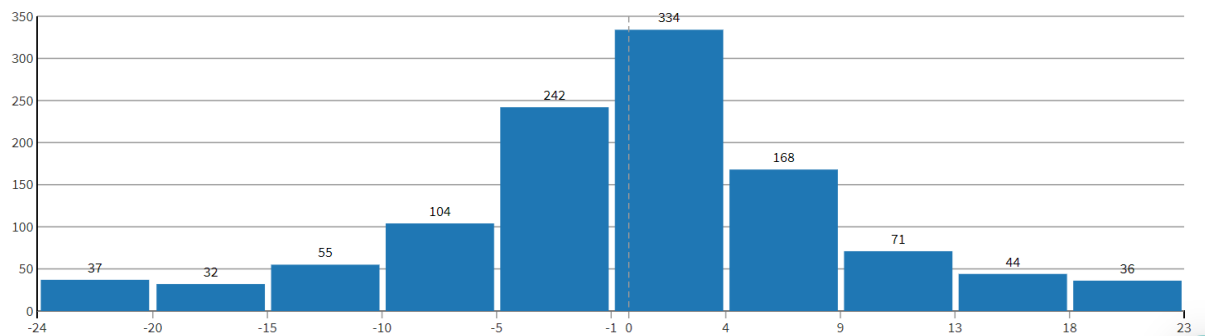
Scatter plot



2.Error Distribution:

Error distribution

Min. (raw)	Min.	25 th perc.	Median	75 th perc.	90 th perc.	Max.	Max. (raw)
-34.403	-24.294	-4.3158	0.37547	5.2048	11.044	22.847	31.766
Average		0.15988		Standard deviation		9.0701	



3.Metrics and Assertion:

Metrics and assertions

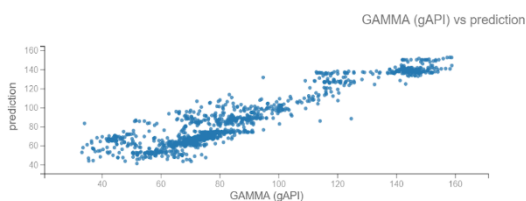
Detailed metrics

Explained Variance Score ?	0.8895
Mean Absolute Error (MAE) ?	6.741
Mean Absolute Percentage Error ?	8.85%
Mean Squared Error (MSE) ?	89.61
Root Mean Squared Error (RMSE) ?	9.466
Root Mean Squared Logarithmic Error (RMSLE) ?	0.1267
Pearson coefficient ?	0.9432
R2 Score ?	0.8895

Model Evaluation:

Model evaluation in machine learning involves assessing the performance and effectiveness of a trained model on unseen data. It helps determine how well the model generalizes to new instances and provides insights into its strengths, weaknesses, and areas for improvement. It's important to select evaluation metrics appropriate for the specific problem domain and the goals of the model.

We use the evaluate recipe provided by dataiku to perform model evaluation. The recipe needs to inputs, one being the input dataset and other being the model that must be evaluated. Here, the input dataset is the test data which contains well data of the well 'C6' and the model that we trained will be the input model. Then we run the evaluate recipe which gives us an output dataset with predicted data.



abs_error_decile

mallint

nteger



Machine learning regression generally involves plotting a line of best fit through the data points. The distance between each point and the line is minimized to achieve the best fit line. As we can see, the gamma value is predicted accurately as the absolute error between the predicted and actual value is in range of 0-2 which is considered negligible. Moreover, from Scatter plot we can also make that a line is formed through the data point which makes an angle of 45 degrees saying that the data reconstructed is accurate. This test data contains well data of a single well, which is not included in the train set. If the model is able to perform well and predict properly for this data, that implies that it would display a similar behaviour with new datasets which need the help of this model.

This Scatter Plot shows the accuracy of the model, which plots a line of best fit through the data points making an angle of 45 degrees.

Regression is a method for understanding the relationship between independent variables or features and a dependent variable or outcome. Outcomes can then be predicted once the relationship between independent and dependent variables has been estimated. It's used as an approach to predict continuous outcomes in predictive modelling, so has utility in forecasting and predicting outcomes from data. Therefore, our model performs its job of reconstructing missing log data accurately. Therefore, this model can be deployed.

Value addition:

Value addition is the value added to the domain by this model. This ML model adds good value to the domain. Log reconstruction manually takes a lot of time and could get laborious depending upon the situation. In some cases, there will be a need to use the equipment and re-collect the whole data which would be an expensive business because of the cost of equipment

used. In scenarios such as reconstructing old log data, or reconstructing log data where one of the feature's values have not been recorded properly or in any other such similar cases it would be a costly affair. In situations like this, the ML model that has been created will be of large use to the organization. Using the other available data, we could train a model that can reconstruct the feature's data using other input features. This model can then be used rather than doing the task manually.

In the above scenario, the regression model will fill the missing data in the target feature column and is evidently reducing the time and effort put into it, as it seems to be highly time taking and strenuous process to perform it manually. Therefore, this model adds high value to the organization.