Information
Technology
Institute

AI-Pro Track – Intake 1
Information Technology Institute
Graduation Project
In

Gamma Ray Log Prediction

Prepared by

Suhaib Ali Hassan

Ahmed Abuabah

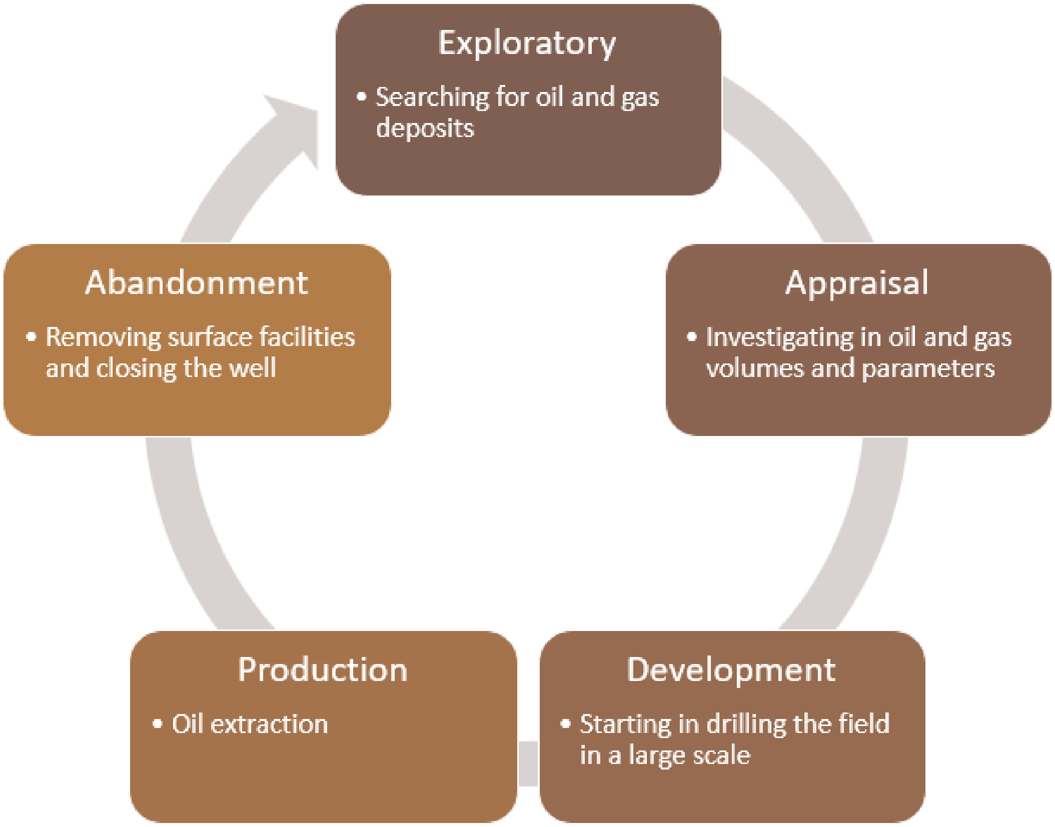
Supervised by
Dr/ Ahmed Fared

2021 – 2022

Contents

1. **Introduction.**

1.1. Well Life Cycle


Exploratory
• Searching for oil and gas deposits
Appraisal
• Investigating in oil and gas volumes and parameters
Development
• Starting in drilling the field in a large scale
Production
• Oil extraction
Abandonment
• Removing surface facilities and closing the well

There are five stages of oil and gas fields' life cycle: **Exploration**, **Appraisal**, **Development**, **Production** and **Abandonment**.

Exploration

Oil and gas exploration involves geological surveys and drilling exploration wells to find viable oil and gas sources to identify areas of potential interest. During the drilling process, general information and samples are collected to know about the rocks, fluids to find out how much oil and gas may be available at the explored area and what is the depth of the oil and gas.

After successful drilling exploration wells,

Appraisal

The main objective is to improve field descriptions through data acquisition, so that the amount of uncertainty about the reservoir size and shape, as well as its marketability, can be reduced.

Development

After successful appraisal and before production. the main objectives are to construct a conceptual development plan to develop the oil and gas field, to design the production wells, to decide which surface and subsurface facilities will be required, and to describe the construction of the facilities and production units.

Production

Once we reach pay zone, the production phase begins. Most oil and gas fields have a lifespan ranging from 15 to 30 years and may be extended up to 50 years or more based on the size of the deposits. After extraction, oil and gas transported for processing and distribution

Abandonment

Wells are plugged and abandoned when oil and gas production is no longer cost effective.Then we reach the end of the oil and gas fields life cycle, and production facilities are removed.

1.2. Logging Mission.

Logging is very important procedure at the well drilling and well completion phases, it acts as our eyes on the well environment properties, so for the decision makers logging is their key-element to take the most suitable optimum decision for that well.

For all this importance Oil companies pay a lot of money for the oil service companies to come with its facilities to measure these logs _ which you may know it now as certain physical measurements that represent the properties of each rock strata over the depth_ directly from the owner companies' wells.

So, our problem here which we are trying to solve is to make sort of cost reduction by predicting one of the most important logs which is very essential log for each well, which is Gamma-Ray Log from the other logs measured for the well through a regression problem using the ML algorithms which we have learned before.

Electric Log
Salt Water Formation
(Conducts Electricity Well)
Oil Formation
(Conducts Electricity Poorly)

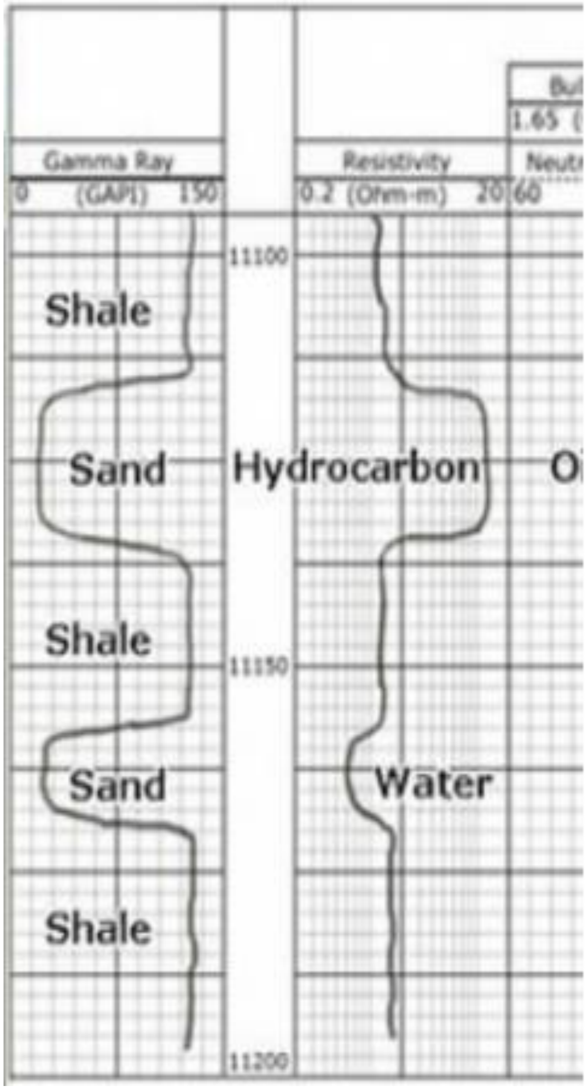
1.3. GR Logging

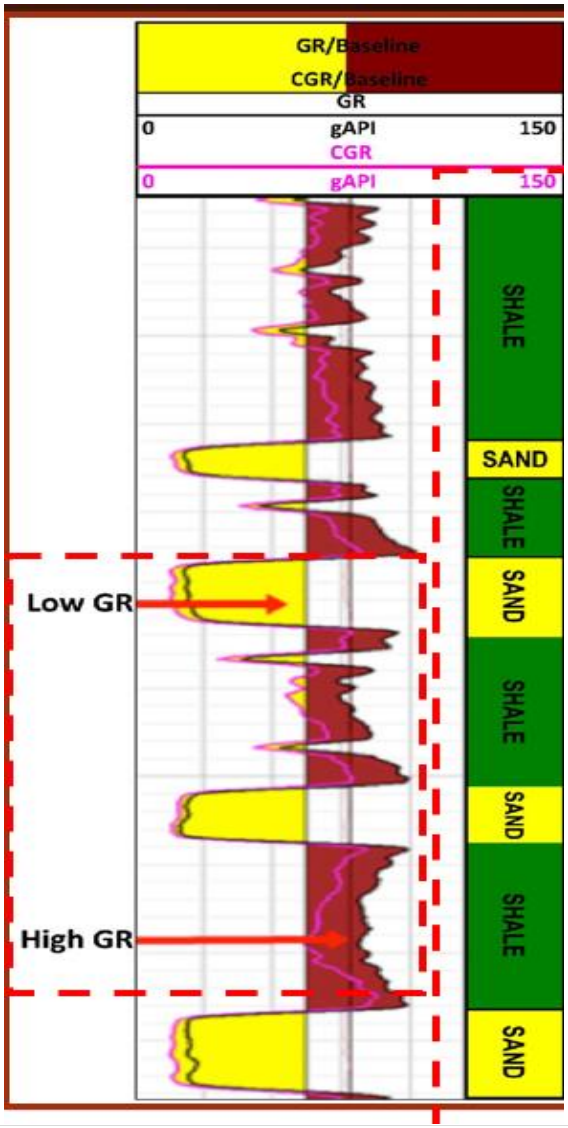
GR log is used to predict the varying lithology in borehole by measuring the spontaneous emission of GR radiation from rocks.

The natural gamma ray (GR, API) log, a necessary logging item in almost all petroleum open holes, measures the total spontaneous radioactivity of the geological formations

record intensity of radioactive source (clay minerals as major component) presented in mineralogical composition of rock section.

The gamma-ray logs are especially useful because shales and sandstones have different gamma-ray signatures, which can be correlated easily among wells.





2. Methodology

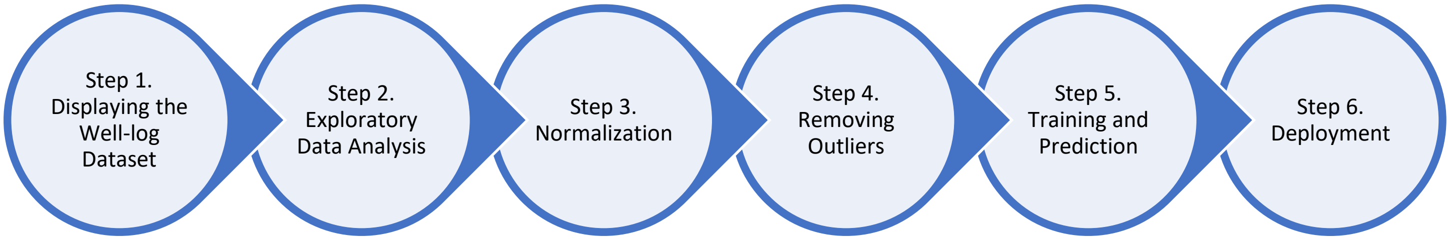
We were provided with a real-data set from our mentor Dr. Ahmed Farid, these well-log datasets have the file format of LAS 2.0, a specific format for well-logs, so it was required to change it into CSV format as to have the ability to read it by python notebooks the working on it.

The goal of this work is to predict the Gamma-Ray Log from the other logs. This will be a supervised learning project. We will train our dataset using regression models. Starting with simple models will be step one in our project. After that, we will apply a more complex model using a neural network model. It will be handled by optimizing hyperparameters to achieve the highest accuracy

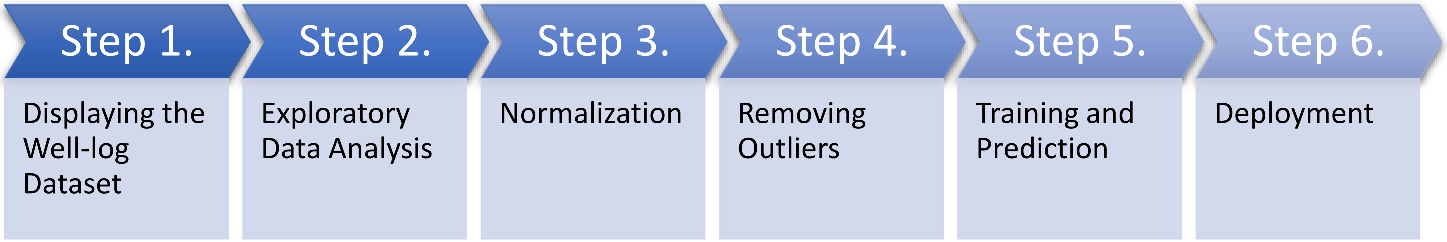
This date represents logging measurements from there different wells from the same field which having these measurements as features all of that is measured versus the MD 'Measured Depth which measured in ft':

- NPHI is the formation porosity, measured in v/v.
- RHOB is the formation bulk density, measured in grams per cubic centimeters.
- PEF is the formation photoelectric absorption factor, dimensionless.
- CALI is the borehole diameter, measured in inch.
- GR is the formation radioactivity content, measured in API "OUR TARGET".

Through the following procedures we hope to reach the lowest RMSE and highest R2 values for this regression problem:


Step 1. Displaying the Well-log Dataset
Step 2. Exploratory Data Analysis
Step 3. Normalization
Step 4. Removing Outliers
Step 5. Training and Prediction
Step 6. Deployment

3. Pipeline



Step 1. Displaying the Well-log Dataset

Step 2. Exploratory Data Analysis

Step 3. Normalization

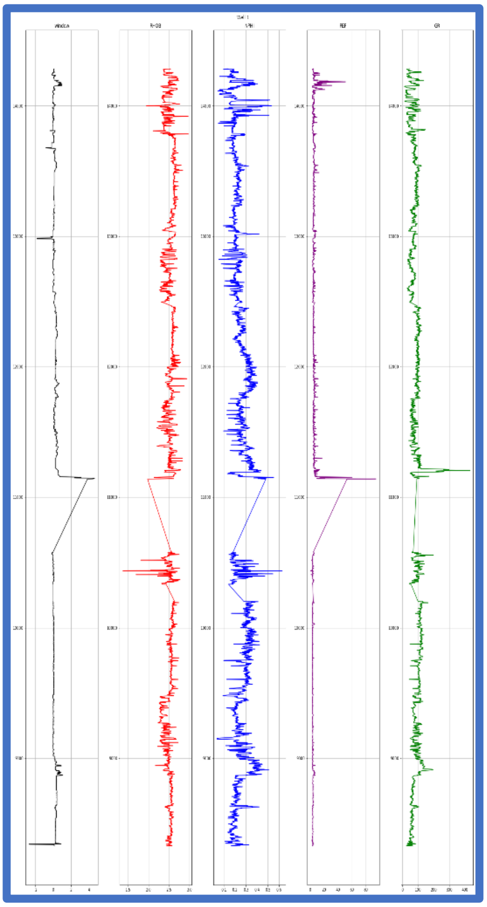
Step 4. Removing Outliers

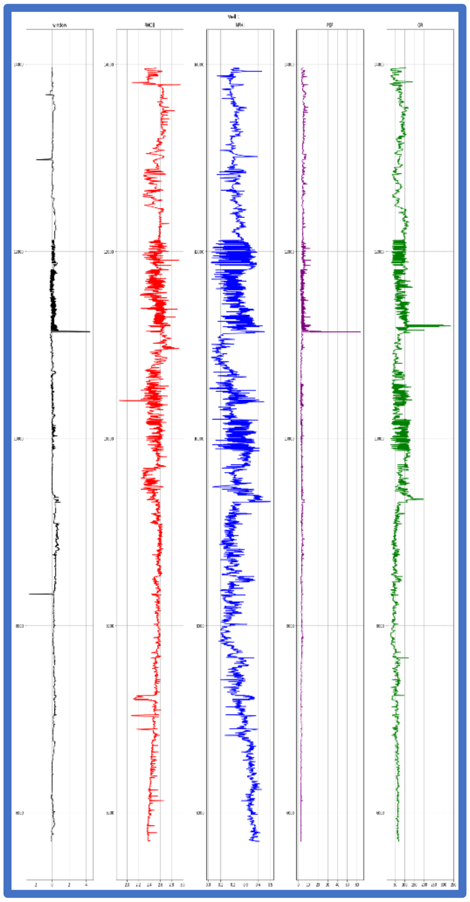
Step 5. Training and Prediction

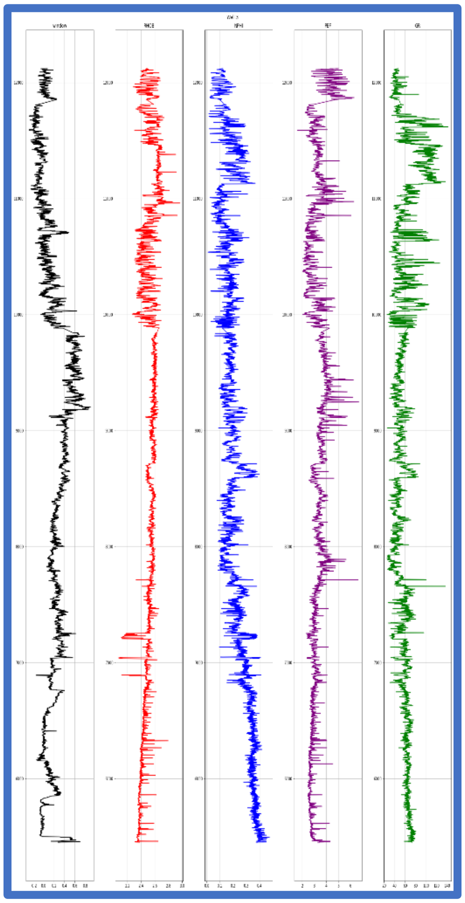
Step 6. Deployment

Step 1. Displaying the Well-log Dataset

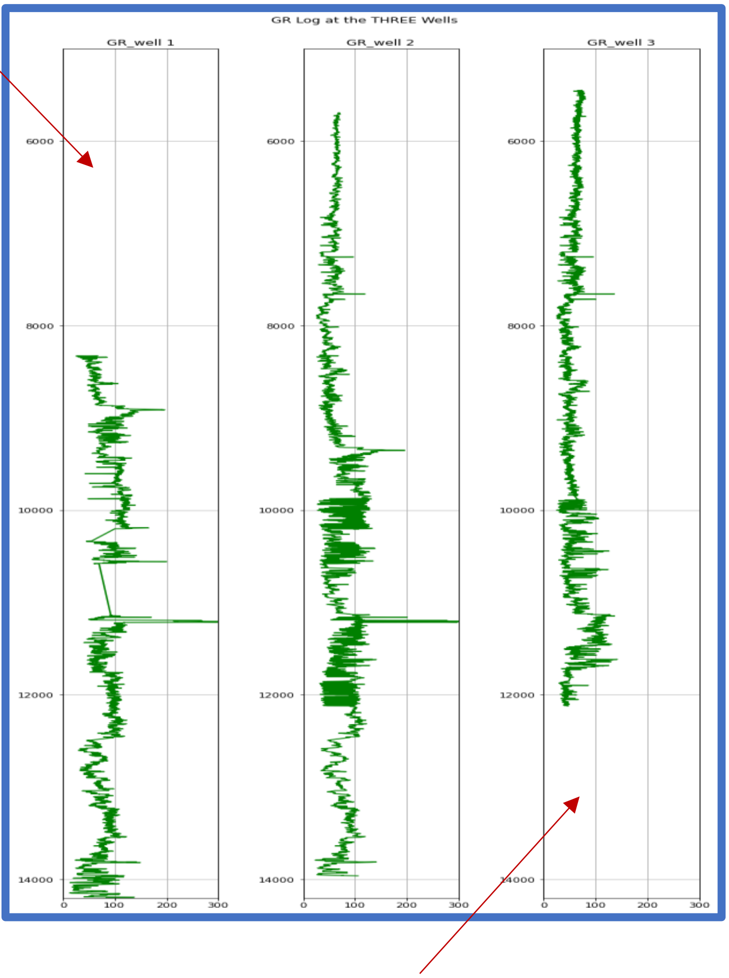
In any formation evaluation, displaying the well-logs is a routine. The following is the display of one of the training data, produced using Matplotlib. We have three plots for each plot represents each log; as we have already discussed, the NPHI, RHOB, GR, RT, PEF, and CALI are the features, and GR is the target.



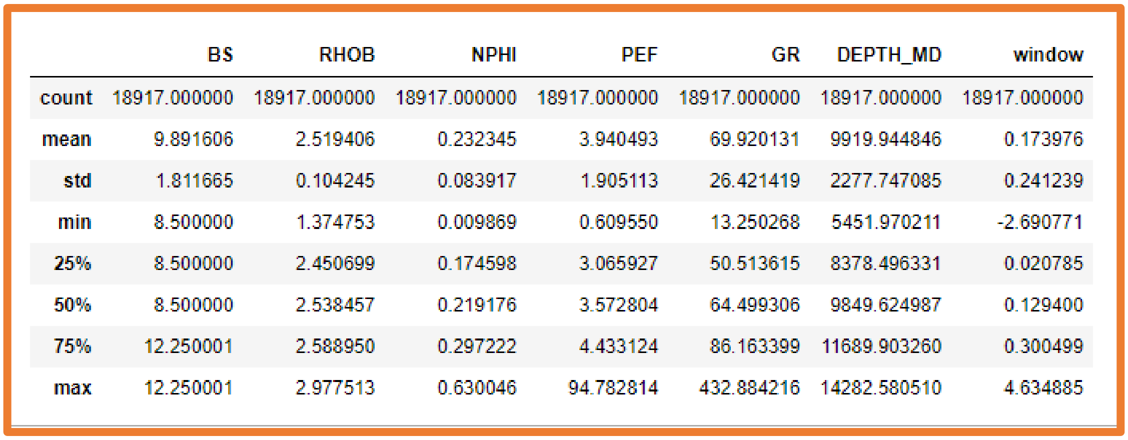




We can see here the benefits of displaying is to make you understand the data well to make your decisions ,so here we can see the depth of every well, for will_1 we have the minimum value of depth = 8251 ft and maximum value 14282 ft, and will_2 we have the minimum value of depth = 5451 ft and maximum value 14282 ft , and will_3 we have the minimum value of depth = 5251 ft and maximum value 12182 ft, so from these values we can decide to make well_1 & well_3 as train data and well_2 as test data, due to the depth of the well so we can see merging well 1 & 3 give us data cover all depth of well_2 to test our model on it.



We have attached some summary data on our wells below:



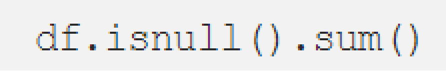
	BS	RHOB	NPHI	PEF	GR	DEPTH_MD	window
count	18917.000000	18917.000000	18917.000000	18917.000000	18917.000000	18917.000000	18917.000000
mean	9.891606	2.519406	0.232345	3.940493	69.920131	9919.944846	0.173976
std	1.811665	0.104245	0.083917	1.905113	26.421419	2277.747085	0.241239
min	8.500000	1.374753	0.009869	0.609550	13.250268	5451.970211	-2.690771
25%	8.500000	2.450699	0.174598	3.065927	50.513615	8378.496331	0.020785
50%	8.500000	2.538457	0.219176	3.572804	64.499306	9849.624987	0.129400
75%	12.250001	2.588950	0.297222	4.433124	86.163399	11689.903260	0.300499
max	12.250001	2.977513	0.630046	94.782814	432.884216	14282.580510	4.634885

Step 2. Data Preparation

Data preparation ensures the accuracy of data, which leads to accurate insights.

First, we need to make sure that our entire data does not contain any **non-numerical values** (NaNs).

we check if NaNs exist in our data.



```
df.isnull().sum()
```

If it turns all zero, we are safe to go. In our case now, it turns all zero and we are already clean from NaNs. Unless so, we need to handle the NaN values.

Next, we **merge the individual well datasets** into two larger single data frames, each for the training and test datasets.

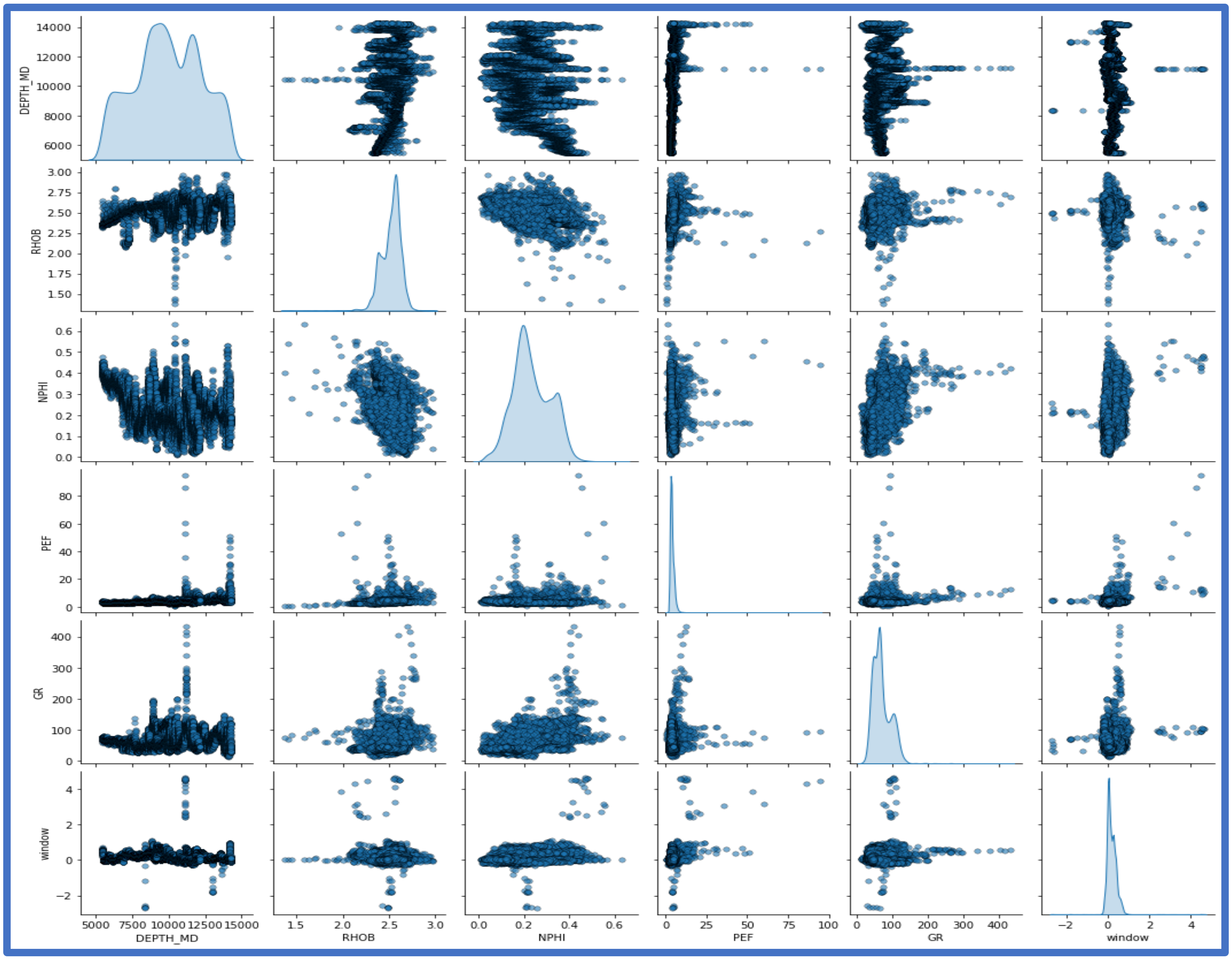
Well_1 & Well_3 will be our training datasets and Well_ will be our test datasets.

```
Train_df = pd.concat([Well_1, Well_3])
```

Step 3. Exploratory Data Analysis

Exploratory data analysis (EDA) is crucial to understand our data. Two important things we want to know is the **distribution** of each individual feature and the **correlation** of one feature to another.

To observe the multivariate distribution, we can use a pair-plot in *Seaborn* package. The following is the multivariate distribution of the features and the target.

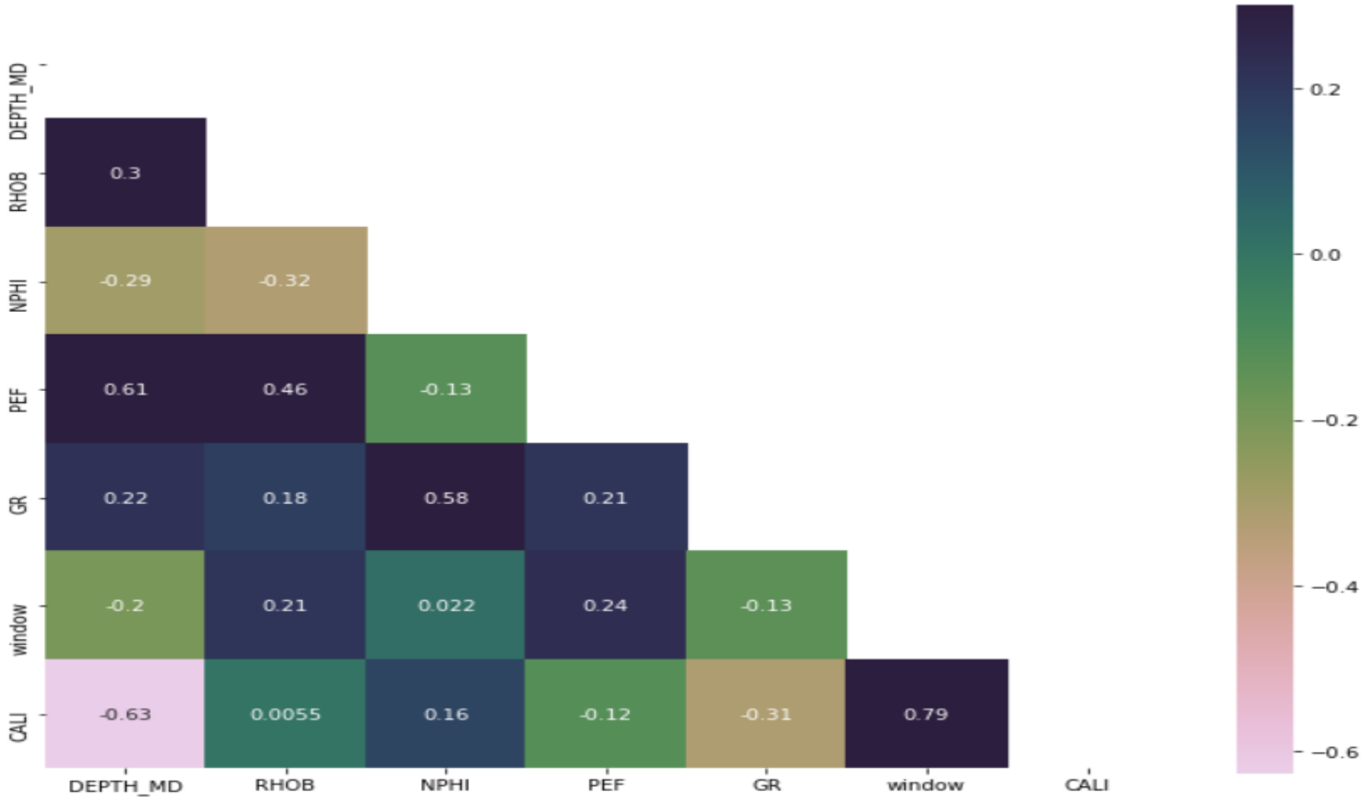


First thing we notice is that most distributions are skewed and not ideally Gaussian, especially the PEF & CALI log. However, for machine learning, Gaussian or less skewed data is preferred. We can do normalization on this data that will be discussed next.

Second, outliers can also be found in the data in the next part , we will discuss how to remove these outliers.

Third, we can see some data pairs are highly correlated, such as NPHI and GR.

We look more into correlations among features and target by calculating the **Spearman's correlation** and visualize the results using the **heatmap**. The following is the Spearman's correlation heatmap of our data.



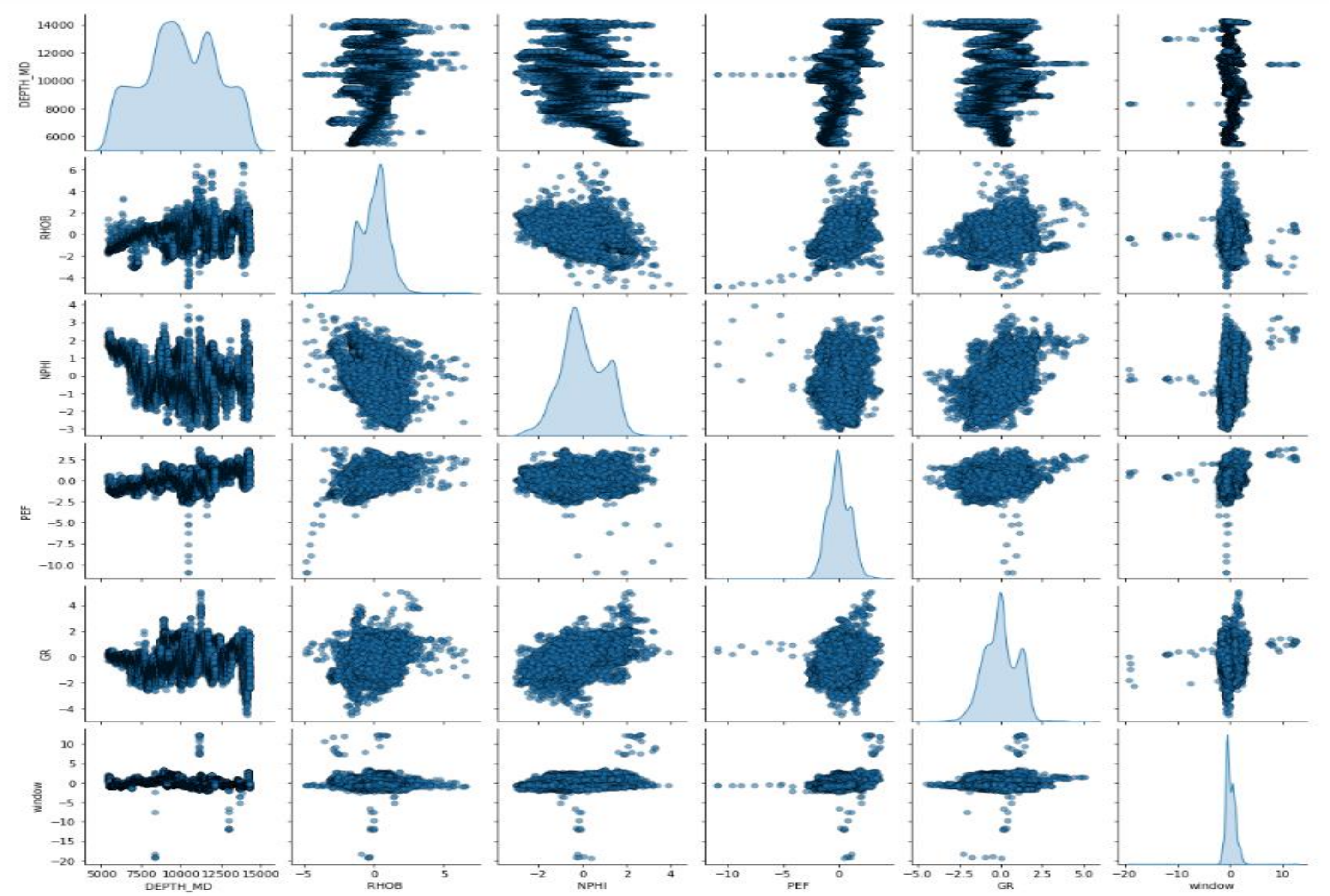
we can obtain here the 2 largest correlations between GR and NPHI (positive correlation of .58) and between GR and PEF (negative correlation of .21). This correlation result matches with the ones we see before in the pair-plot. Also, RHOB seems to have little correlation with other features.

As a common practice, any feature with very low correlation is excluded for prediction, hence CALI can be excluded.

Step 4. Normalization

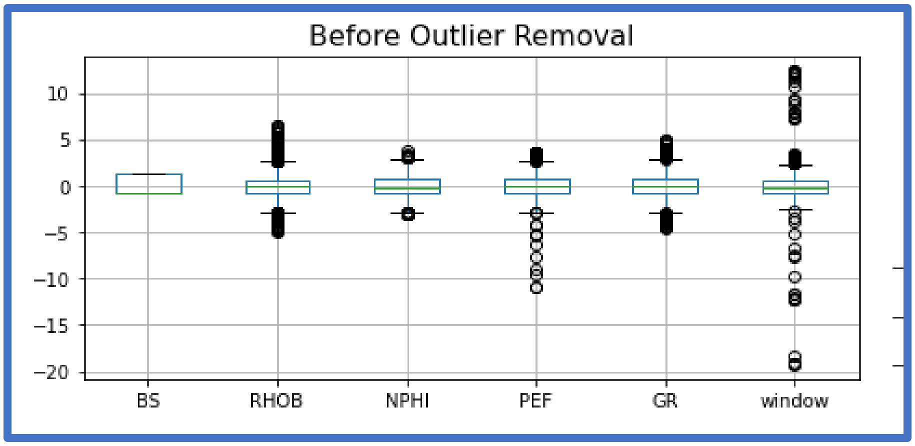
As we know from the pair-plot, most distributions appear skewed. In order to improve our prediction performance, later on, we'd better do a normalization. Normalization is a technique to transform the data (without changing it) to be better distributed.
The widely used normalization techniques are **standardization** (by transforming with the mean and standard deviation) and **min-max** (with minimum and maximum value).

After trying all the methods, we find the power transform technique is the best technique using Yeo-Johnson method the most powerful method with our data.



Step 5. Removing Outliers

We found a significant number of outliers during our analysis, and the box plot below clearly illustrates this.

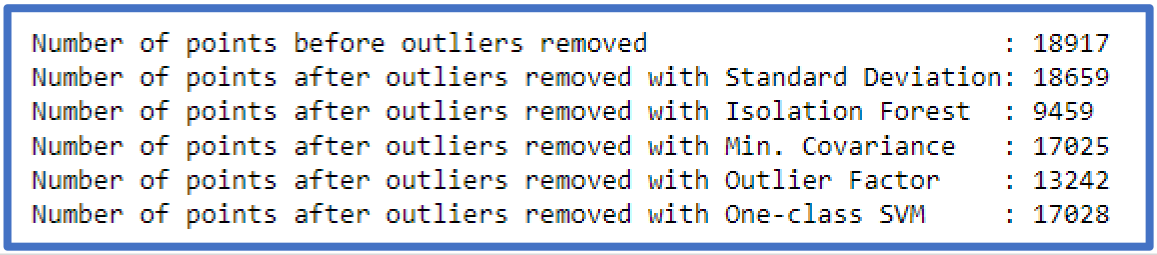


We remove outliers because their presence can affect prediction performance by using five methods.

Standard deviation removal is the most widely used method, which is also the most basic. In this method, we specify the threshold as the min and max values away from the standard deviation. We can build this ourselves.
Scikit-Learn provides several outlier removal methods, such as **Isolation Forest**, **Minimum Covariance using Elliptic Envelope**, **Local Outlier Factor**, and **One-class Support Vector Machine**.

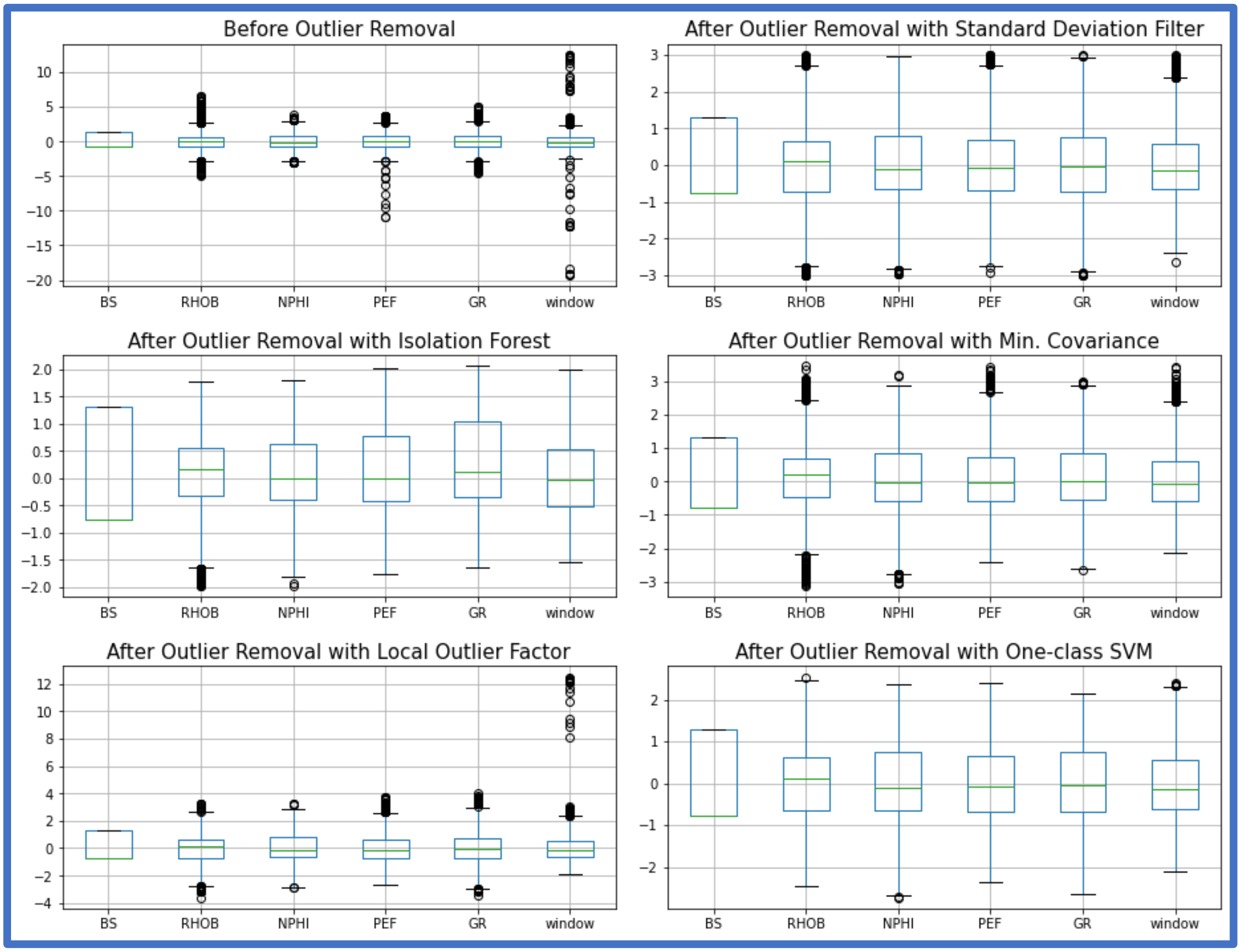
Each of the five methods are implemented.
In order to compare which removal method is the best, I use two methods.

First, for each method, it's possible to count the data before and after the outliers are removed.



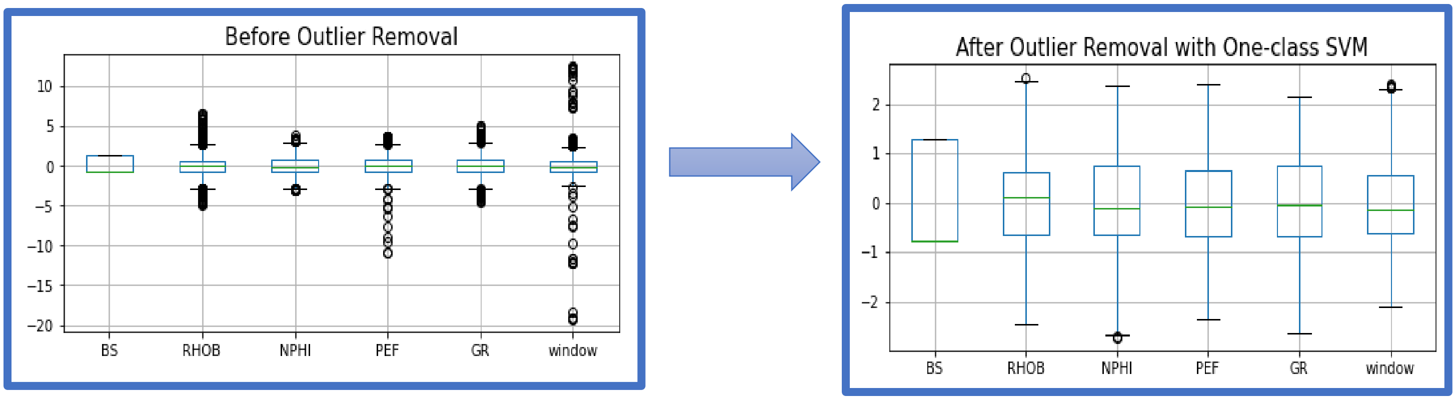
```
Number of points before outliers removed                   : 18917
Number of points after outliers removed with Standard Deviation: 18659
Number of points after outliers removed with Isolation Forest  : 9459
Number of points after outliers removed with Min. Covariance   : 17025
Number of points after outliers removed with Outlier Factor    : 13242
Number of points after outliers removed with One-class SVM     : 17028
```

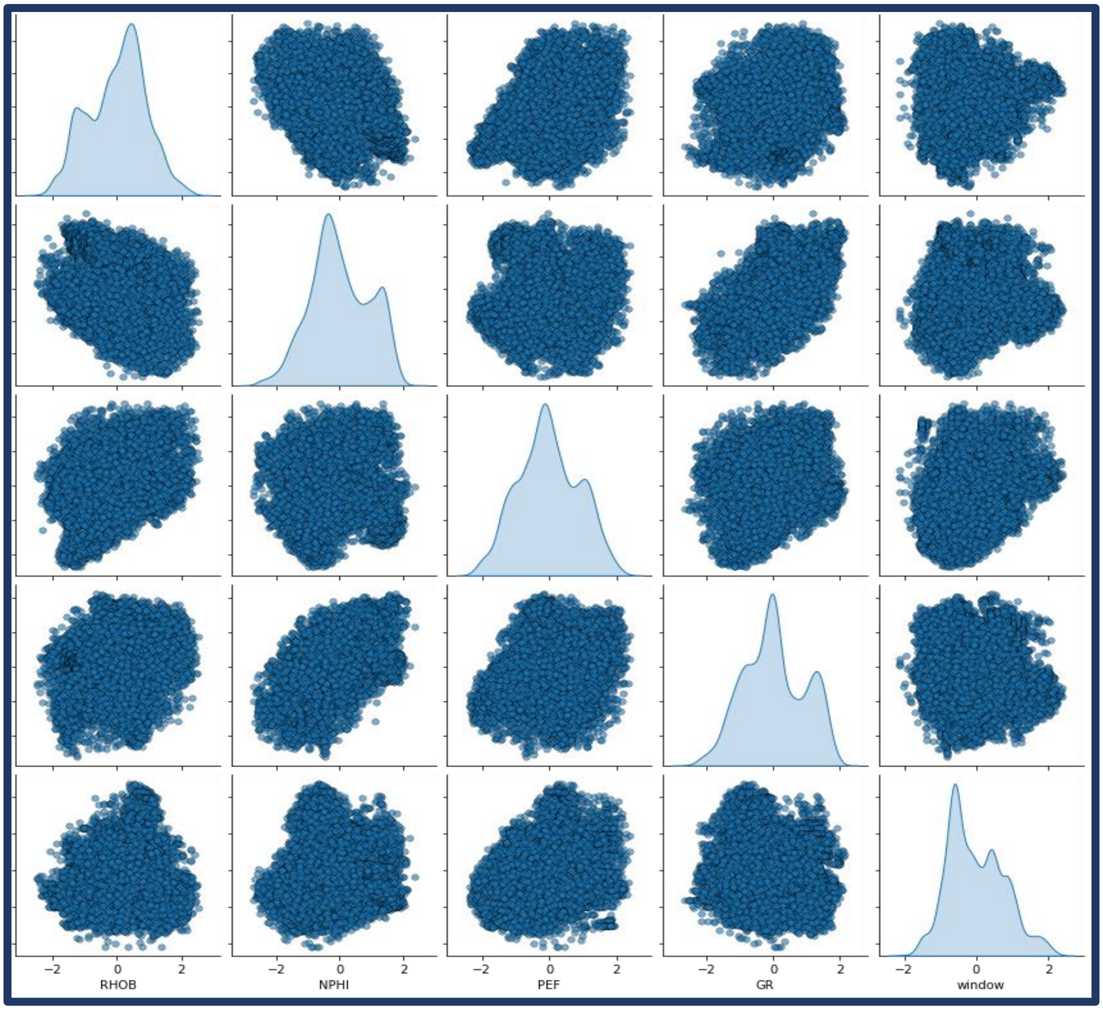
Then, to decide which one is better between all outliers removal methods, I produce the boxplots for each feature before and after normalization. Below are the boxplots.



Now it becomes visible by observing the number of outliers that **One-class SVM performs "cleaner"** any other methods. Although Isolation Forest is also clean, but still One-class SVM is the winner.

The conclusion is: we use **One-class SVM**. Again, we produce a pair-plot to observe the final result of our data.



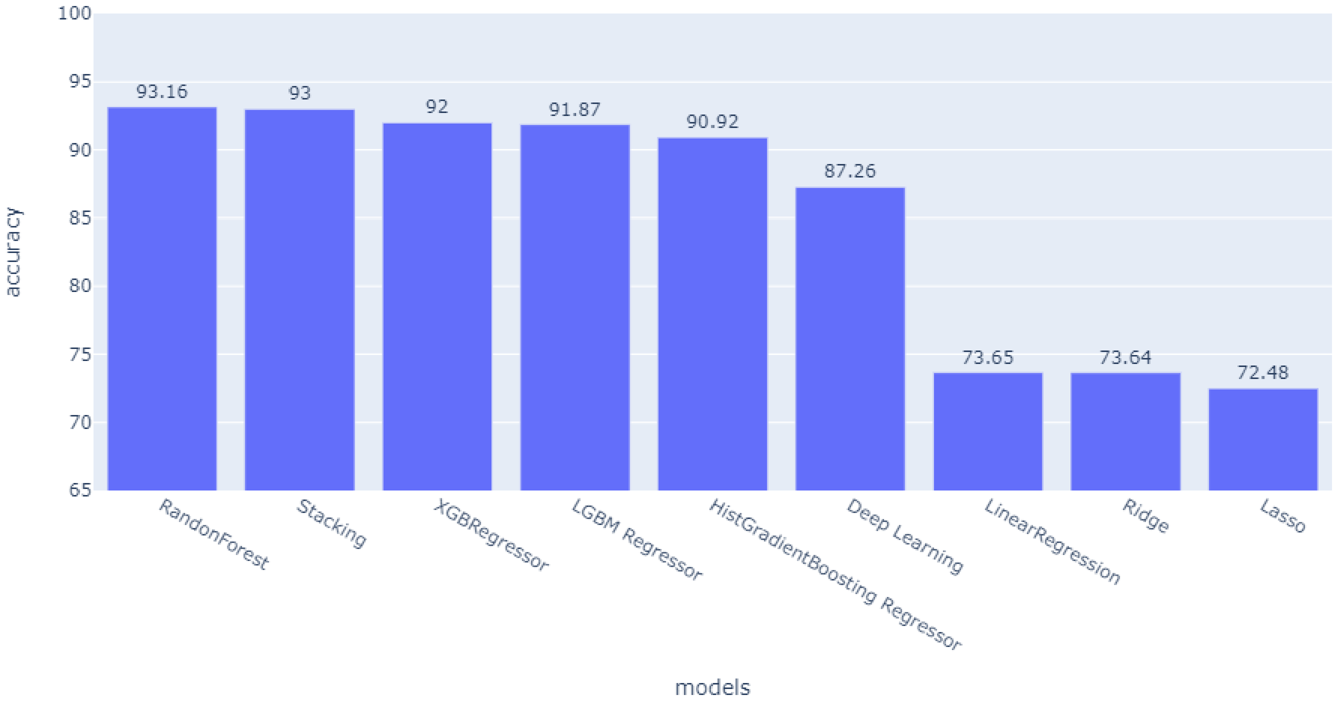


Step 5. Training and Prediction

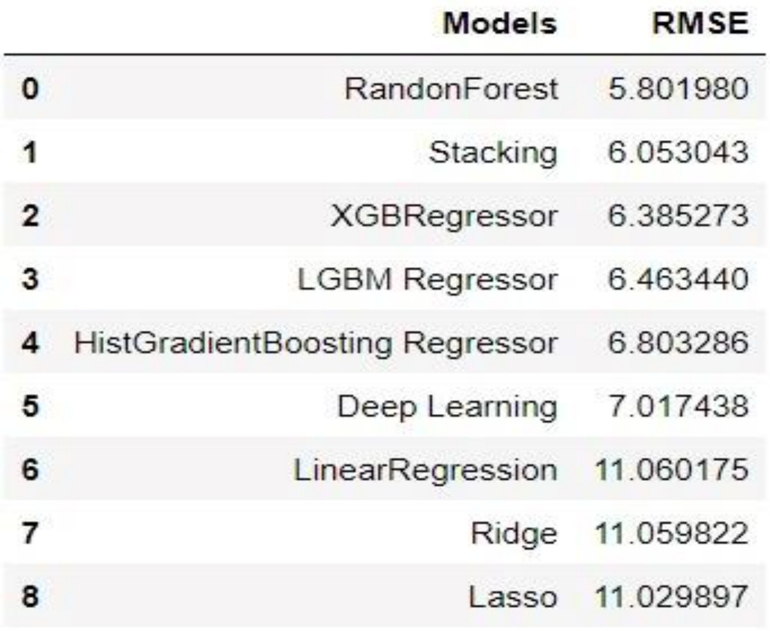
In the fifth step, we have not yet actually do the prediction to our real test data. We need to evaluate the performance of each regression model that we use by **training the train data and testing the model to the train data themselves**, and then we evaluate how close the predicted GR log to the true GR log.

For training and prediction, we used variable models e.g. (Random Forest, Stacking, XGB Regressor, LGBM Regressor, HistGradientBoosting Regressor, Deep Learning, Linear Regression, Ridge, Lasso) for the quick run and made Hyperparameter Tuning for the promising one for our data set.

I use R^2 and root mean squared error (RMSE) as the scoring metric to measure the performance of each regression model. The following is the result of the R^2 score for each regressor.



The following is the result of the root mean squared error (RMSE) for each regressor.



	Models	RMSE
0	RandonForest	5.801980
1	Stacking	6.053043
2	XGBRegressor	6.385273
3	LGBM Regressor	6.463440
4	HistGradientBoosting Regressor	6.803286
5	Deep Learning	7.017438
6	LinearRegression	11.060175
7	Ridge	11.059822
8	Lasso	11.029897

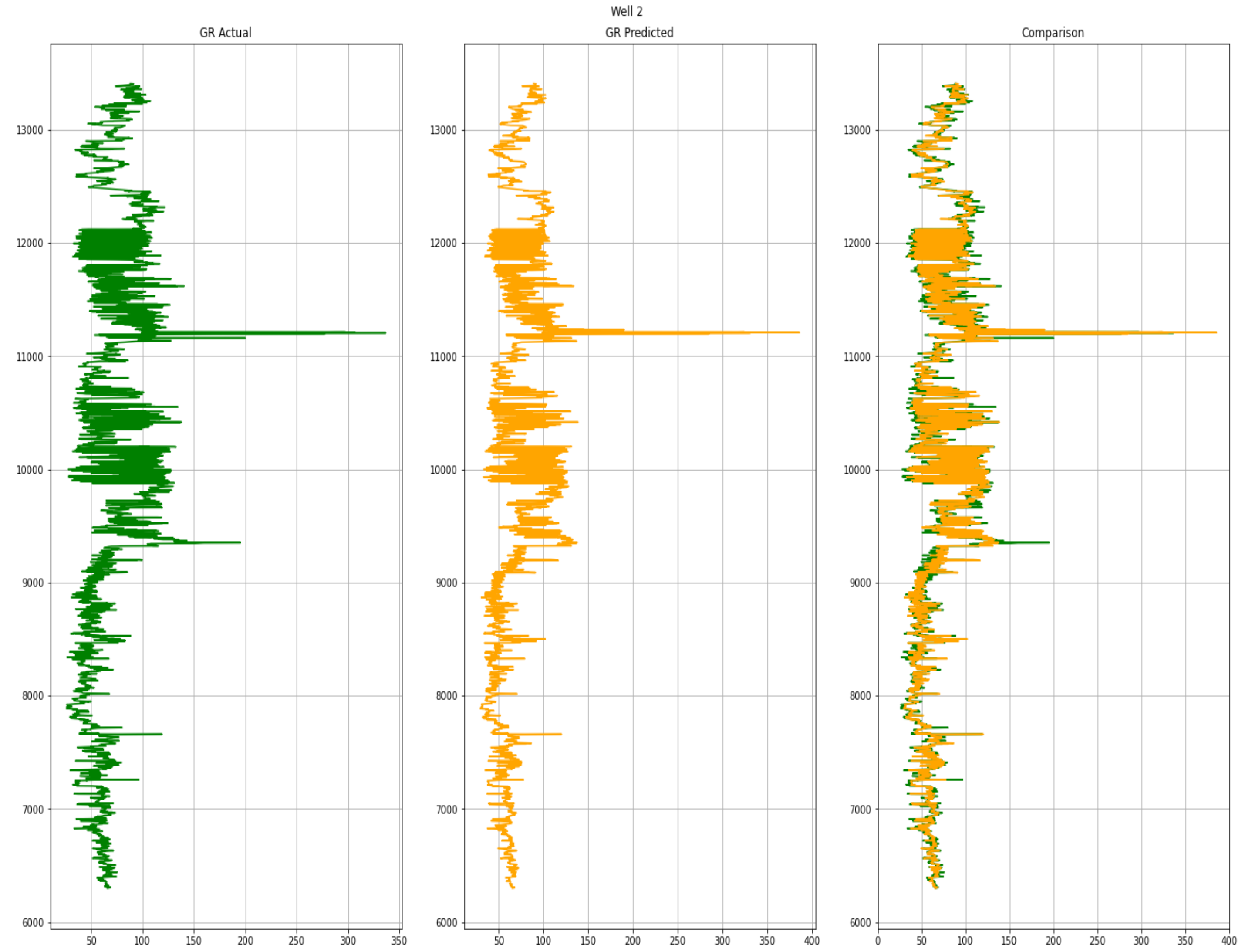
As we can see we find the **Random Forest Model** is the highest model in accuracy, and lowest in RMSE so we used it in the Deployment phase.

4. Result

The following are the predicted GR log and the actual GR log in well 2 with accuracy 93.1 and RMSE 5.8.

As it is obvious from the plots below, both predicted and actual GR logs are very close.

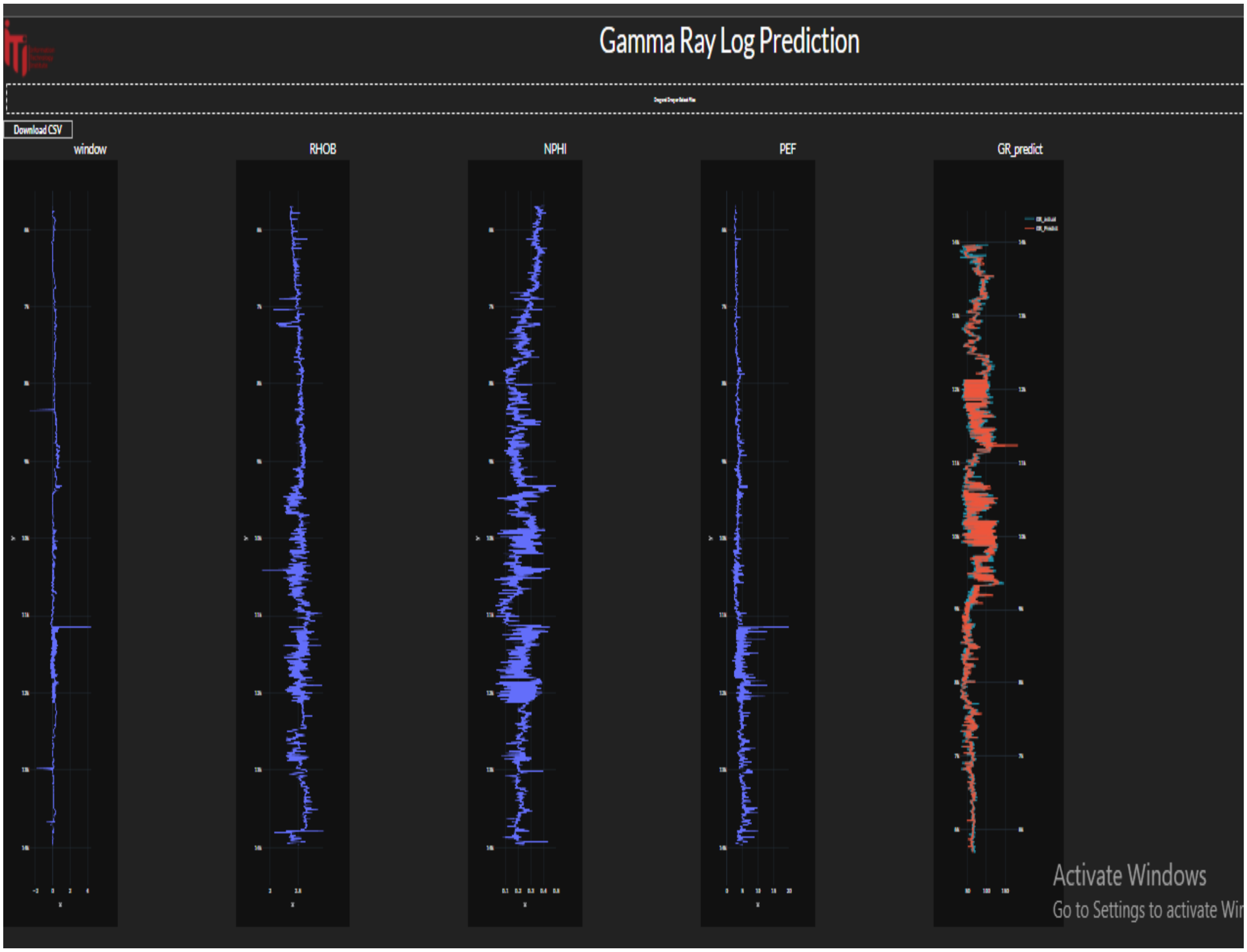
The last plot shows how closely the predicted GR log and the actual GR log match, and this is very apparent in the comparison between the two.



5. Deployment

Finally, we reach the end of our journey, we have deployed our model in a way that anyone can use it.

By using Herokuapp we can deploy our model, and anyone can upload a data file to provide a plot of all the logs and predicted GR log as shown below.



Here is our deployment link attached below :

https://gammarayprediction.herokuapp.com/

Here is our code on Github:

https://github.com/Suhaib000/Gamma_Ray_Log_predicion

What is Next???

Future Work of The Project:

- Contacting with the service companies to integrate the model with their tools' software.
- Deploying the model with more data from the owner companies after presenting the model to them seeking for testing and enhancing the model.
- Rebuilding the model but in a different type of data sets such as Drilling parameter logs (WOB, ROP, SPP, T, …….).
- Publishing a paper with our results at the project to document what happened.

THANKS ALOT