Chemical Plant Data Analysis

Exercise E11

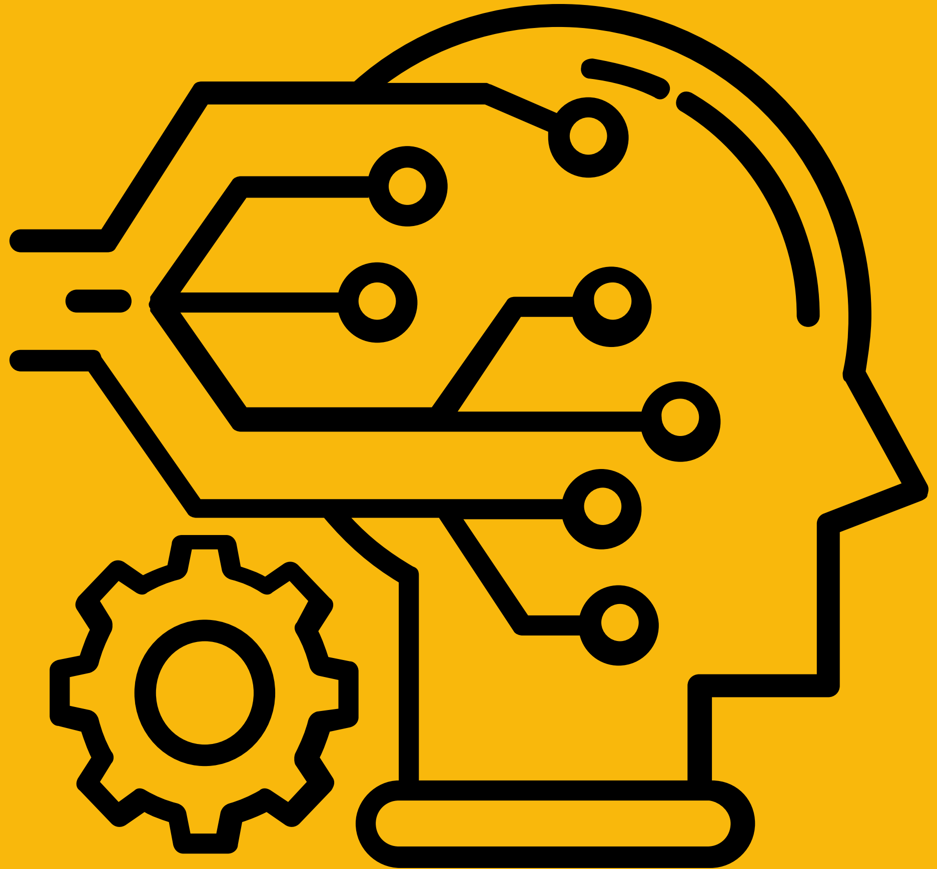
Kshitij Vaidya : 22B1829
Pinak Mahapatra : 22B0447
Suyash Gahankari : 22B0426

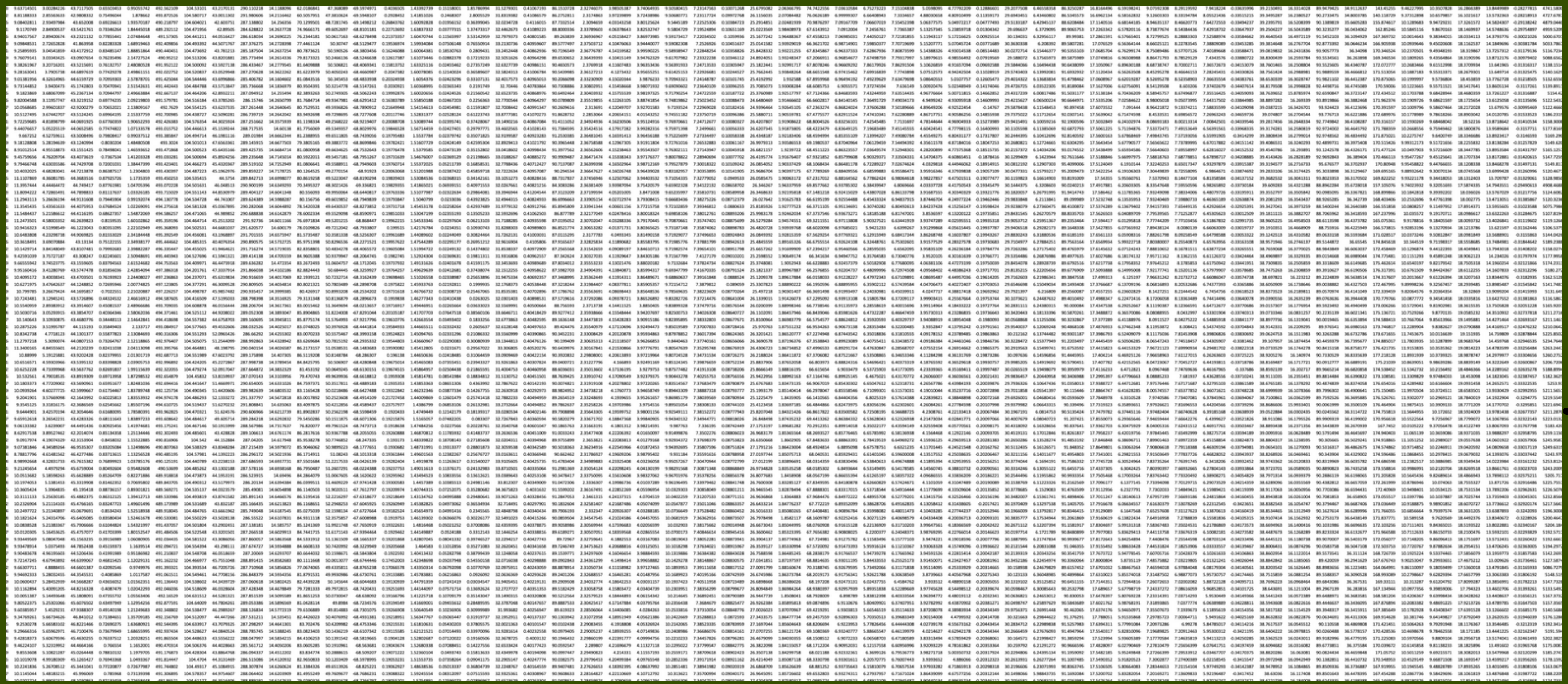
Problem Statement

- The data we are provided is acquired from the system of a chemical processing plant comprising 241 columns and 1025 rows
- We have been given the controllable parameters and the ones to be monitered
- The parameters to be monitered contain the vibration data of certain critical equipment which MUST be kept under control
- Also understand Specific Energy - energy consumed per unit output

Tasks to Perform

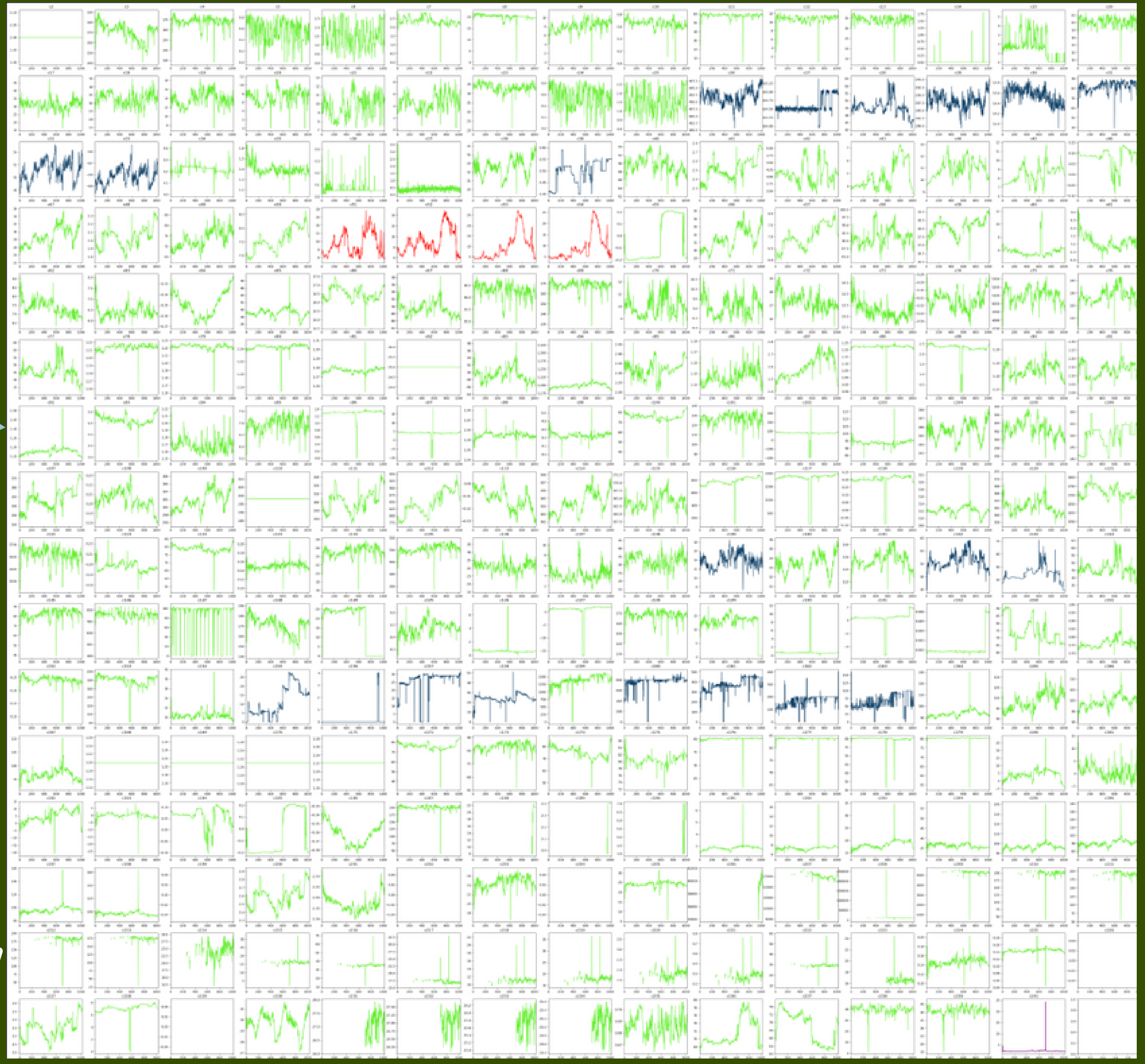
- Create ML models to predict the vibration levels based on the operating and controllable parameters
- Furthur, create ML models using ONLY the controllable parameters to predict and control the vibrations
- Analyse the parameters affecting Specific Energy and find the minimum number of independent parameters that can be used to predict their values
- Our goal is to create a predictive model based on these independent variables




[illegible]

An Excel Snapshot of Our Data Acquisition

SCALE OF OUR DATASET!!

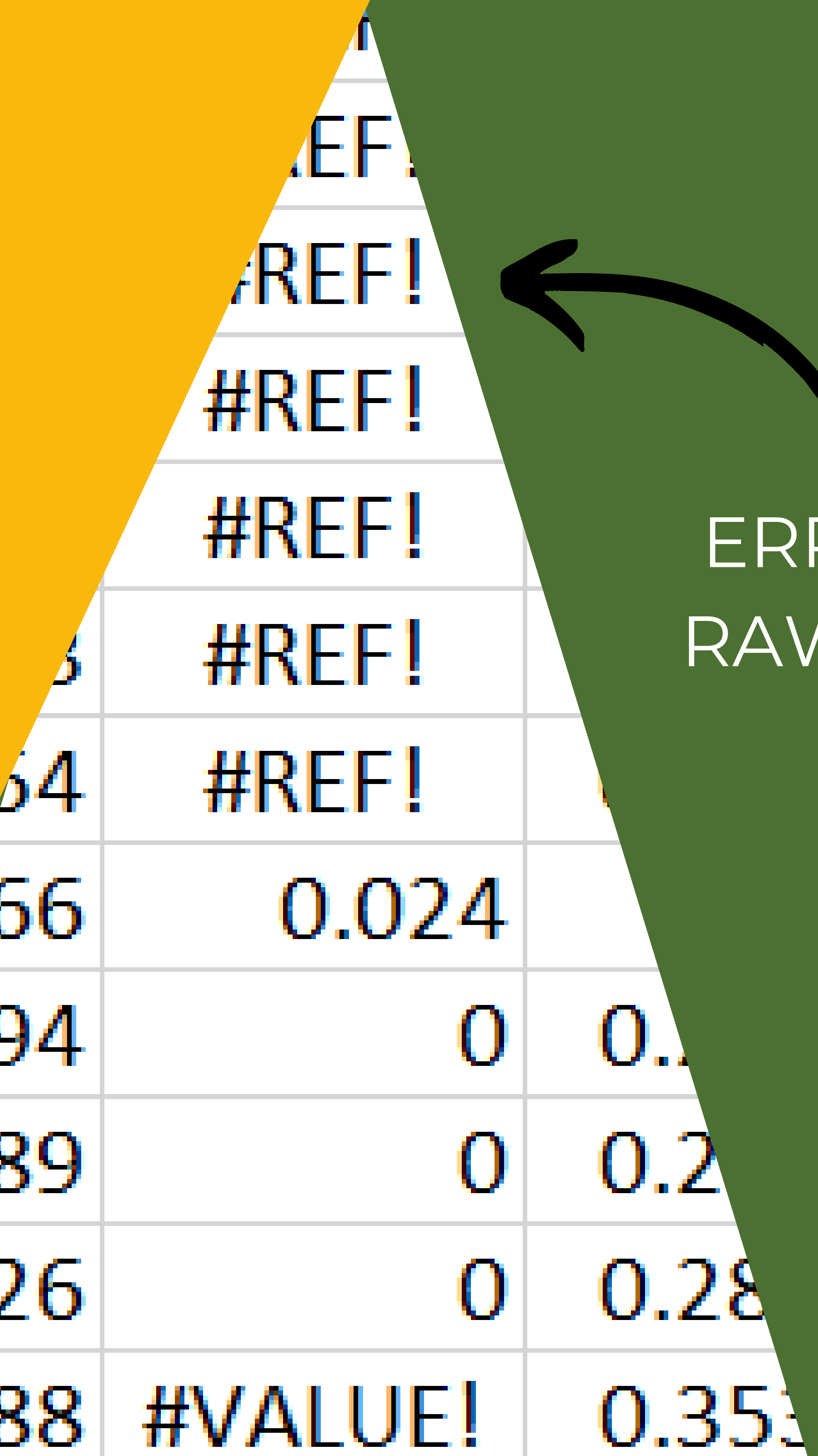


Plotting our Parameters Columnwise-
Green:Operating, Dark Blue:Controllable, Red: Monitor,
Light Blue: Specific Energy

Major Challenges Faced ...

- The major challenges encountered in our data analysis and machine learning endeavors revolved around data cleaning and preprocessing.
- Missing data posed a common hurdle, necessitating decisions on whether to eliminate incomplete records or employ advanced techniques like interpolation. Outliers, with their potential to skew results, demanded careful identification and consideration for removal or transformation.
- Ensuring consistency in data formats and handling duplicate records were critical for unbiased analyses. Transforming data to align with algorithmic assumptions, and strategically engineering features, emerged as key steps to bolster model performance.

EXPLORATORY DATA ANALYSIS

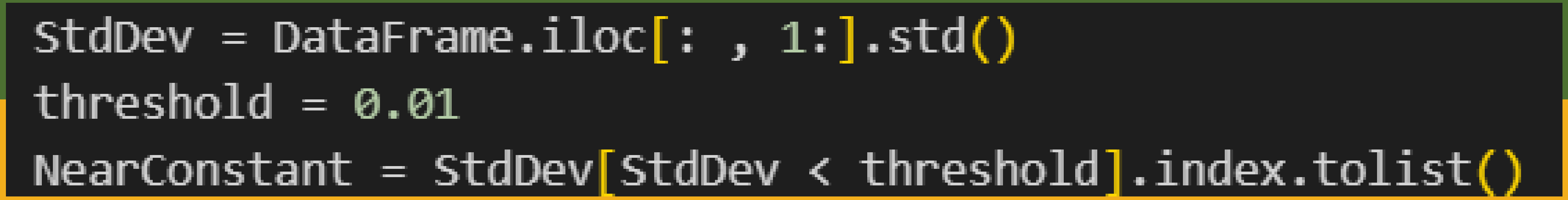


ERROR IN RAW DATA

- In the initial stages of data cleaning and preprocessing, we start by identifying and handling erroneous data points, denoted by entries such as '#REF', '#VALUE', and '#DIV/0'. These are replaced with NaN values to uniformly deal with all the undesired value types in the future.
- The data is then organized into distinct sections, including controllable, monitor, specific energy, and operating parameters, facilitating a more granular understanding of how these variables influence the overall dataset

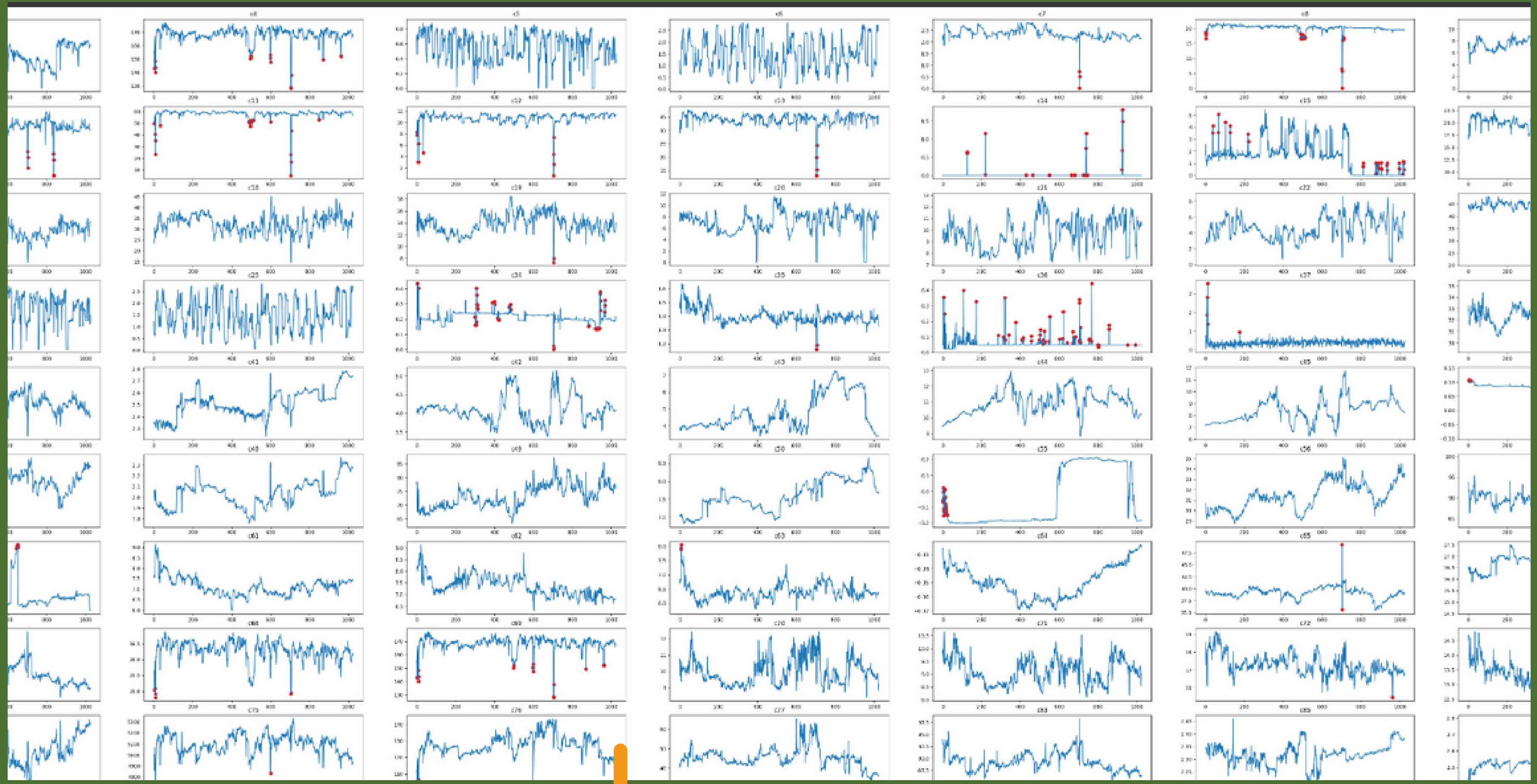
CLEANING THE DATA: Handling NaN Values and Constant Value Columns

- We first calculate the count of NaN values in each column/parameter of the dataset.
- We observe that all the parameters consisting of NaN values lie in the Operating Parameters. Owing to the large size of dataset we can drop all the columns with atleast one NaN value.
- We then identify the columns in Operating Parameters with constant values by calculating their standard deviation and setting a threshold value of 0.01 and because of their constant nature we drop these columns also.
- Total columns dropped= 71



```
StdDev = DataFrame.iloc[: , 1:].std()
threshold = 0.01
NearConstant = StdDev[StdDev < threshold].index.tolist()
```

DEALING WITH OUTLIERS

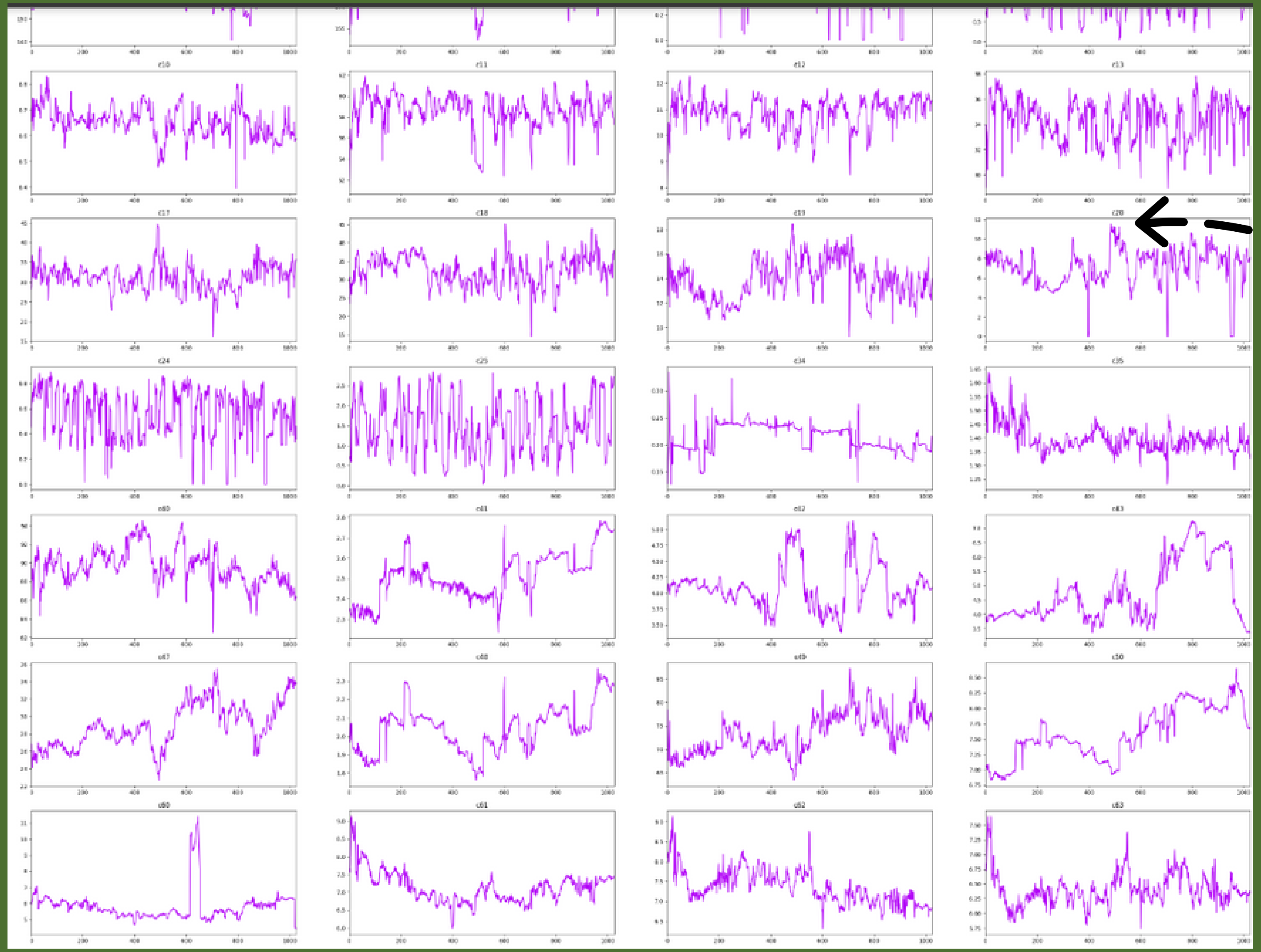


DATA SET WITH OUTLIERS

- We apply a rolling window approach to identify local outliers in each column of the time series dataset. The outliers are stored in a list with each element corresponding to a column in the original dataset.
- To refine outlier detection using the interquartile range (IQR), we partitioned the dataset into segments and applied IQR-based thresholding individually. This localized approach revealed previously overshadowed outliers for more accurate identification.
- Range- 15th to 85th percentile
  Threshold= 2*IQR

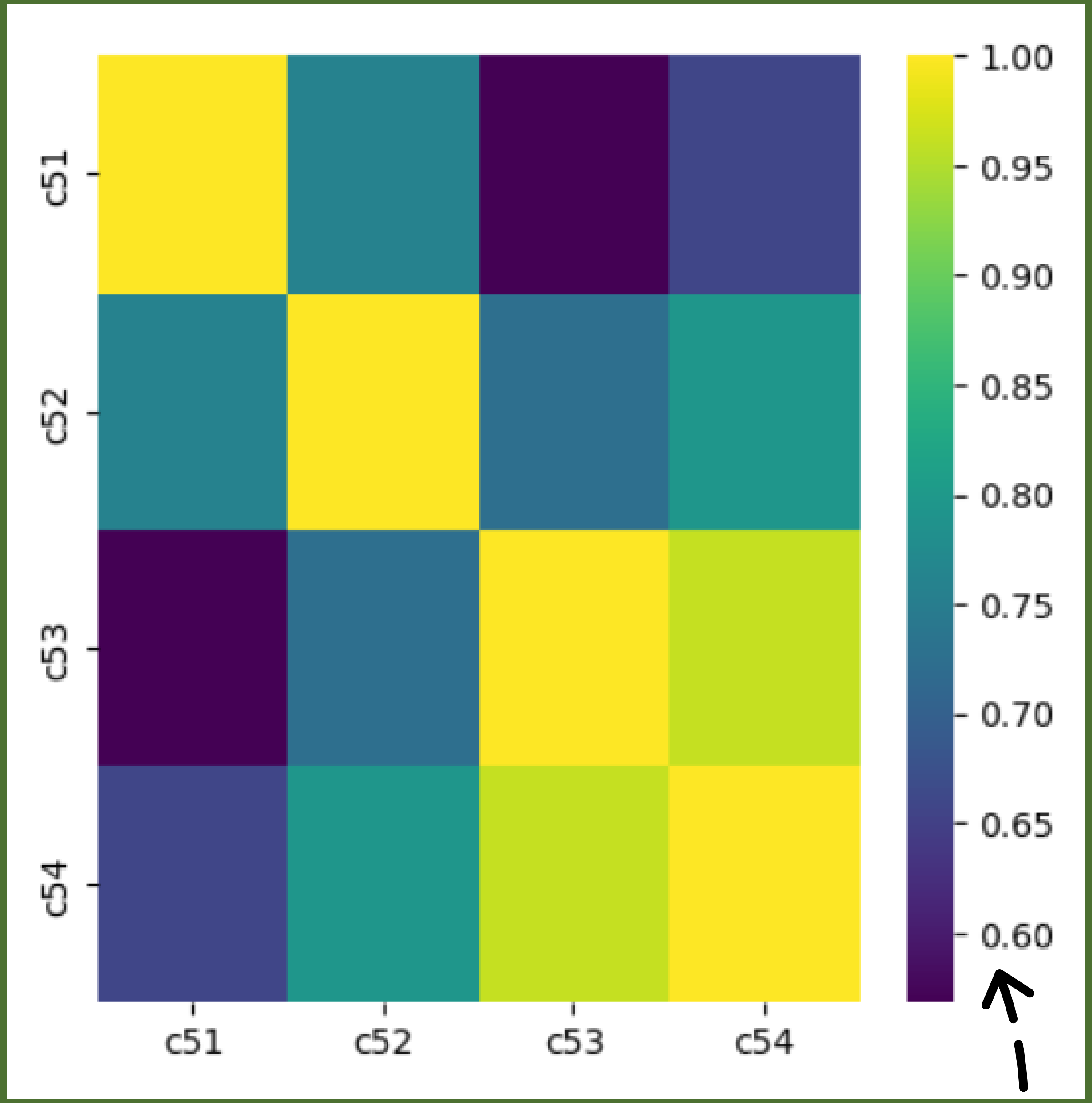
FIXING THE OUTLIERS!

Our approach loops through columns in the remaining data set, dropping outlier rows using stored indices. Afterwards, missing values in each column are filled using linear and nearest-neighbour interpolation, enhancing data quality for analysis or modelling.



DATA SET AFTER FIXING OUTLIERS

Correlation between the vibration vartiables

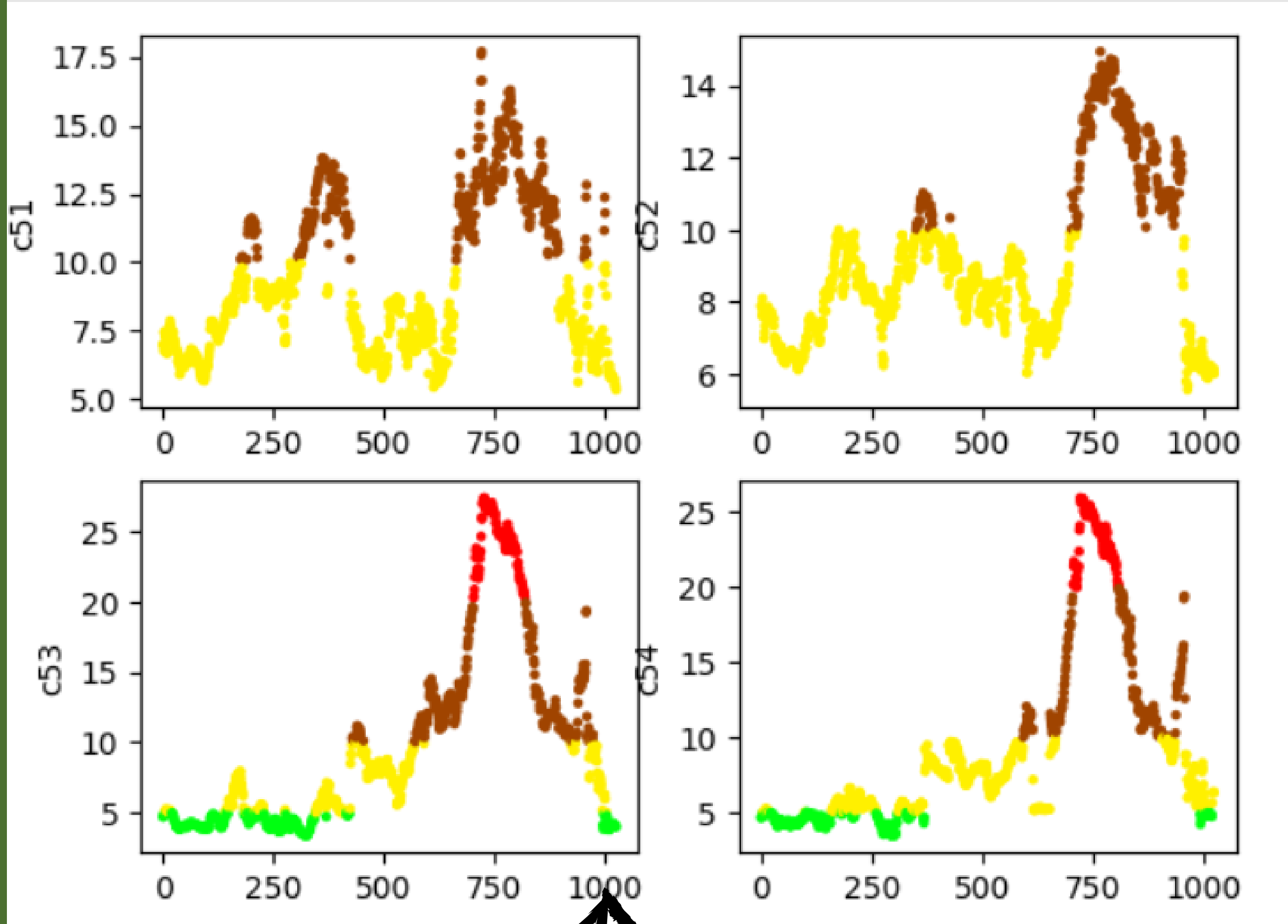


HEAT MAP

- Heatmaps are a valuable tool in exploratory data analysis and data interpretation, offering an effective means to visually represent intricate numerical data in an intuitive and easily understandable format. They excel in revealing patterns, trends, and relationships within datasets.
- The provided heatmap illustrates the relationships among the variables c51, c52, c53, and c54. The values are color-coded based on the scheme presented alongside the heatmap.

VIBRATION READING CATEGORIZATION!

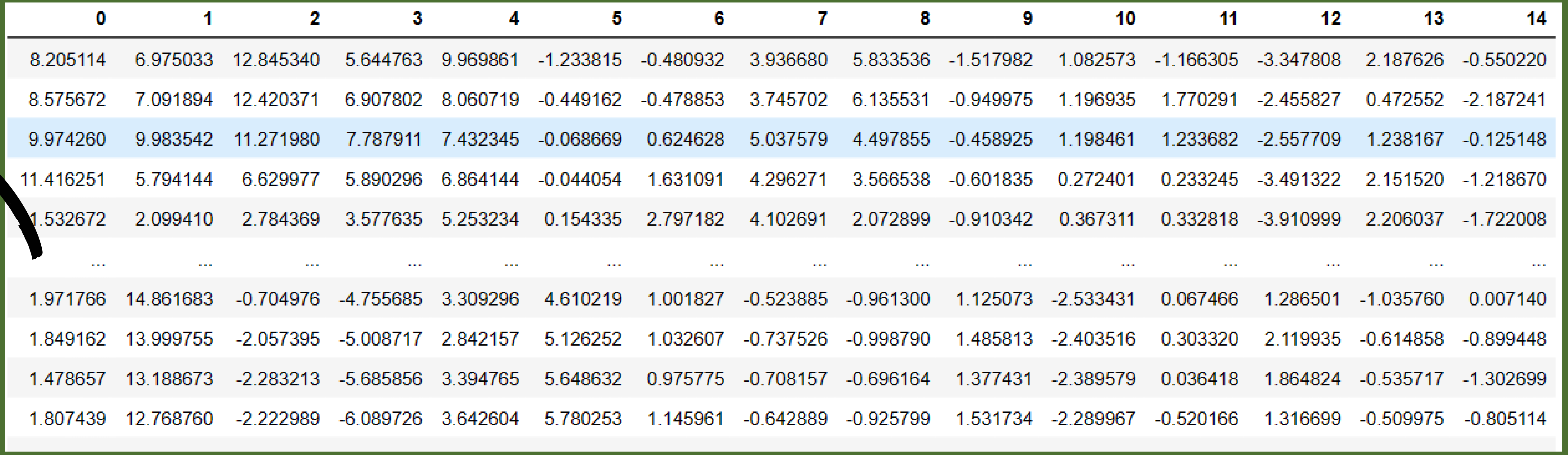
- We defined a `StatusCheck` function, which utilizes the nested `Segregate` function to categorize readings into status levels based on predefined thresholds. The resulting status values are applied to specific columns in a `Moniter Data` Data Frame.
- Next, the `ColourSeparate` function assigns status-based colors to scatter plots for 'c51', 'c52', 'c53', and 'c54', providing a brief overview of their status and magnitude.



SUBPLOTS SHOWING SAFTEY LEVELS

1. ML Model using Controllable and Operating Parameters To Predict Vibration Levels

Set of Final 15 variables



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	8.205114	6.975033	12.845340	5.644763	9.969861	-1.233815	-0.480932	3.936680	5.833536	-1.517982	1.082573	-1.166305	-3.347808	2.187626	-0.550220
	8.575672	7.091894	12.420371	6.907802	8.060719	-0.449162	-0.478853	3.745702	6.135531	-0.949975	1.196935	1.770291	-2.455827	0.472552	-2.187241
	9.974260	9.983542	11.271980	7.787911	7.432345	-0.068669	0.624628	5.037579	4.497855	-0.458925	1.198461	1.233682	-2.557709	1.238167	-0.125148
	11.416251	5.794144	6.629977	5.890296	6.864144	-0.044054	1.631091	4.296271	3.566538	-0.601835	0.272401	0.233245	-3.491322	2.151520	-1.218670
	1.532672	2.099410	2.784369	3.577635	5.253234	0.154335	2.797182	4.102691	2.072899	-0.910342	0.367311	0.332818	-3.910999	2.206037	-1.722008
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	1.971766	14.861683	-0.704976	-4.755685	3.309296	4.610219	1.001827	-0.523885	-0.961300	1.125073	-2.533431	0.067466	1.286501	-1.035760	0.007140
	1.849162	13.999755	-2.057395	-5.008717	2.842157	5.126252	1.032607	-0.737526	-0.998790	1.485813	-2.403516	0.303320	2.119935	-0.614858	-0.899448
	1.478657	13.188673	-2.283213	-5.685856	3.394765	5.648632	0.975775	-0.708157	-0.696164	1.377431	-2.389579	0.036418	1.864824	-0.535717	-1.302699
	1.807439	12.768760	-2.222989	-6.089726	3.642604	5.780253	1.145961	-0.642889	-0.925799	1.531734	-2.289967	-0.520166	1.316699	-0.509975	-0.805114

FIRST ML MODEL: DIMENSIONALITY REDUCTION

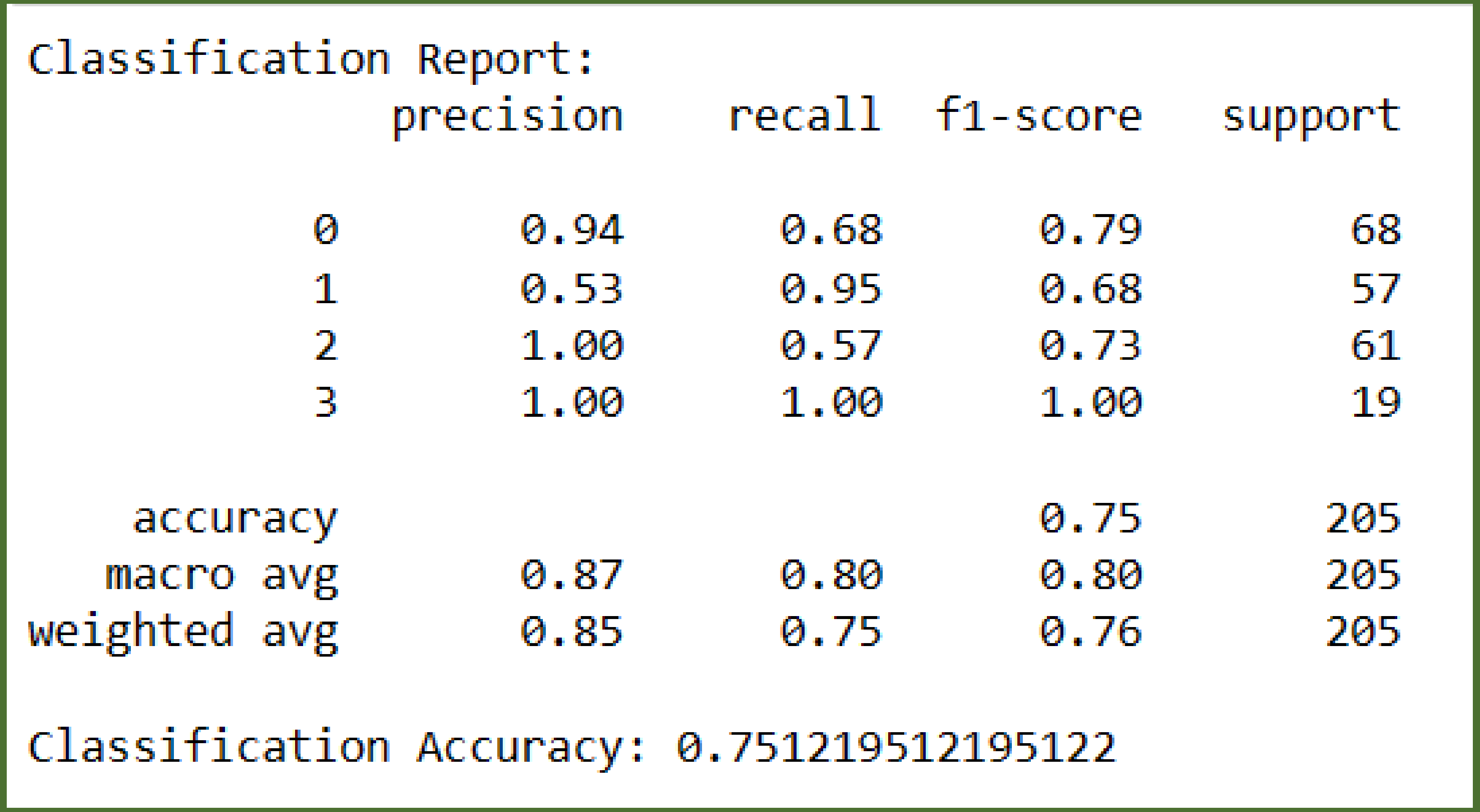
- One challenge we faced when we tried to apply regresison was to minimize the number of variables(Controlling and Operating Parameters).
- Utilizing scikit-Learn's PCA, we successfully reduced the regression model's independent variables to 15, preserving key features and achieving robust results in subsequent regressors and classifiers.
- Using Standard scaler and RandomForestRegressor on above dataset our model effectively predicts vibration levels.

TRAINING AND TESTING THE DATA

- We split the data into training and testing sets to effectively evaluate the performance of our model. Our model performed exceptionally well on the training data with an R^2 value of 0.994 and a Mean Squared Error of 0.55 .
- This performance was nearly replicated on the testing dataset(signs of no overfitting) as well showing an R^2 value of 0.960 .
- These metrics highlight the performance of our model and its ability to make very accurate predictions about the required data.

CLASSIFICATION REPORT!

- We set class thresholds for 'c53' as [5, 10, 20]. Predicted and true values were discretized into classes using `np.digitize`.
- Using scikit-learn's `accuracy_score`, we assessed the model's classification accuracy, presenting detailed metrics in a classification report. The overall accuracy provides a comprehensive evaluation of model performance.

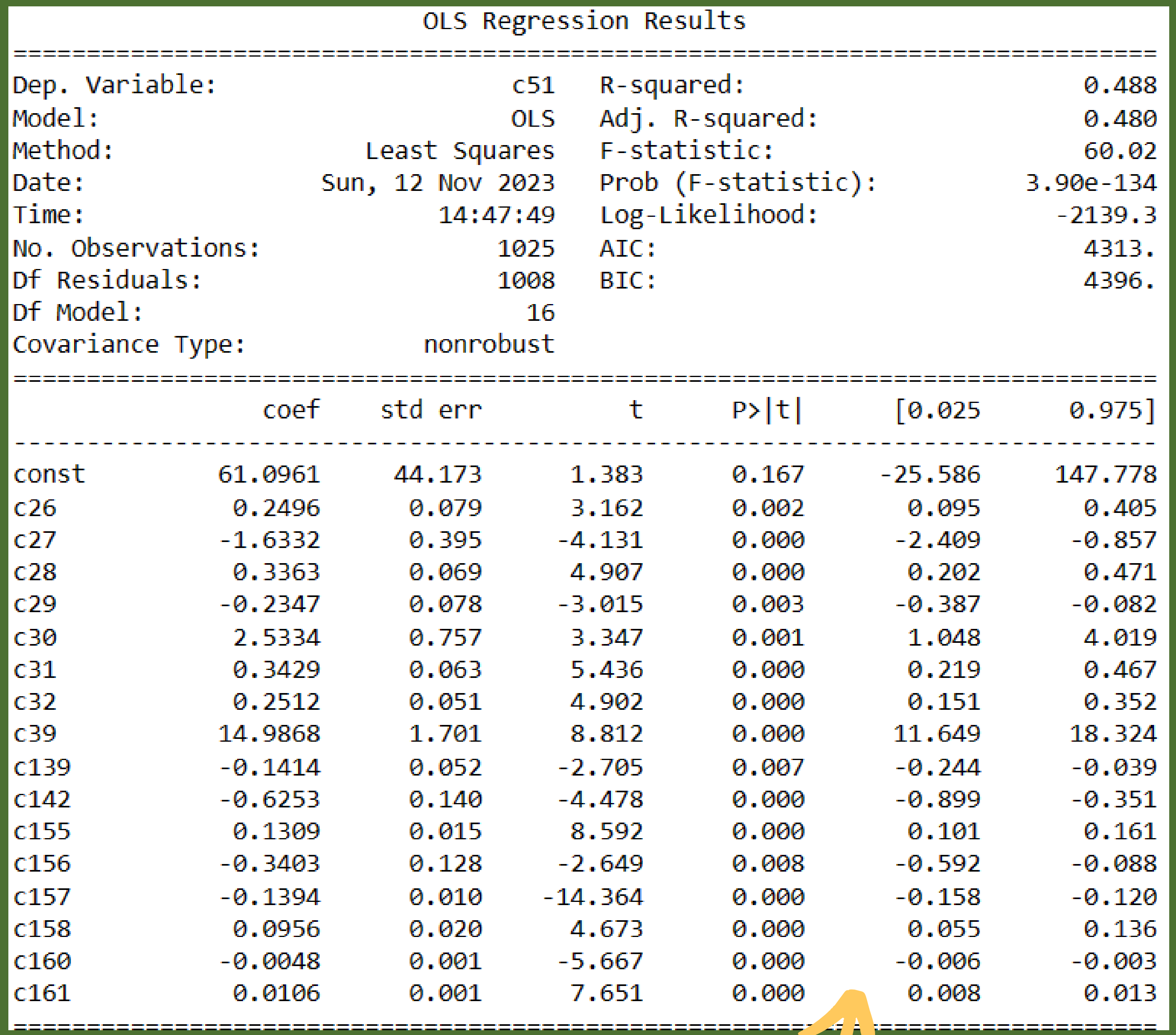


```
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.68      0.79        68
           1       0.53      0.95      0.68        57
           2       1.00      0.57      0.73        61
           3       1.00      1.00      1.00        19

    accuracy                           0.75       205
   macro avg       0.87      0.80      0.80       205
weighted avg       0.85      0.75      0.76       205

Classification Accuracy: 0.751219512195122
```

2. ML model to control vibrations using only Controllable Parameters

Controlling and Monitoring c51 Column

- Using Multiple Linear Regression and OLS model from stats models, we predicted and controlled 'c51' vibrations. After training on Controllable Data, we refined the model by removing parameters with P-values > 0.05, resulting in a final model with 16 predictors, excluding c162, c33, c163, and c143.
- In the Final MLR Model, key parameters (c39, c30, c27, c142) significantly influence controlling 'c51' vibrations.
- In the heatmap, 'c51' showed weak correlations. Our MLR model, with an impressive R^2 of 0.960, suits 'c51' prediction but not necessarily other parameters due to weaker correlations.

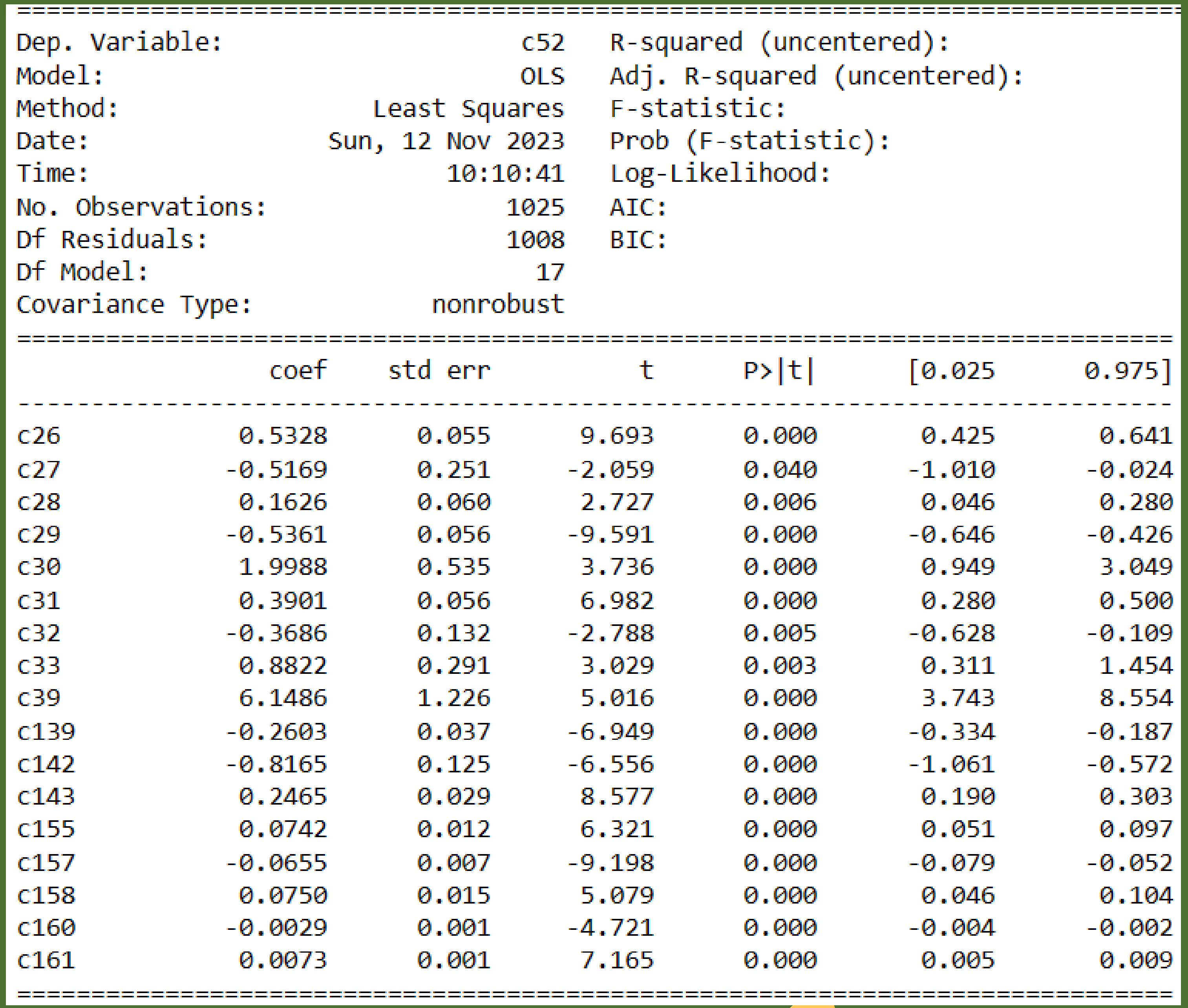


```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    c51   R-squared:                       0.488
Model:                            OLS   Adj. R-squared:                  0.480
Method:                 Least Squares   F-statistic:                     60.02
Date:                Sun, 12 Nov 2023   Prob (F-statistic):          3.90e-134
Time:                        14:47:49   Log-Likelihood:                -2139.3
No. Observations:                1025   AIC:                             4313.
Df Residuals:                    1008   BIC:                             4396.
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         61.0961     44.173      1.383      0.167     -25.586     147.778
c26            0.2496      0.079      3.162      0.002       0.095       0.405
c27           -1.6332      0.395     -4.131      0.000      -2.409      -0.857
c28            0.3363      0.069      4.907      0.000       0.202       0.471
c29           -0.2347      0.078     -3.015      0.003      -0.387      -0.082
c30            2.5334      0.757      3.347      0.001       1.048       4.019
c31            0.3429      0.063      5.436      0.000       0.219       0.467
c32            0.2512      0.051      4.902      0.000       0.151       0.352
c39           14.9868      1.701      8.812      0.000      11.649      18.324
c139          -0.1414      0.052     -2.705      0.007      -0.244      -0.039
c142          -0.6253      0.140     -4.478      0.000      -0.899      -0.351
c155           0.1309      0.015      8.592      0.000       0.101       0.161
c156          -0.3403      0.128     -2.649      0.008      -0.592      -0.088
c157          -0.1394      0.010    -14.364      0.000      -0.158      -0.120
c158           0.0956      0.020      4.673      0.000       0.055       0.136
c160          -0.0048      0.001     -5.667      0.000      -0.006      -0.003
c161           0.0106      0.001      7.651      0.000       0.008       0.013
==============================================================================
```

Summary Results!

Controlling and Monitoring c52 Column

- Analyzing 'c52,' we used OLS with ControllableData, starting with 20 parameters. Iteratively dropping features based on MLR Model P-values, we excluded c162, c163, and c156.
- Post this we obtained a 17-feature MLR model emphasizing c39, c30, c33, and c142's significance in controlling 'c52.'
- The correlation heatmap indicates low correlation of 'c52' with other vibration parameters, limiting the model's applicability to control them. The final model achieves an impressive R^2 value of 0.978.

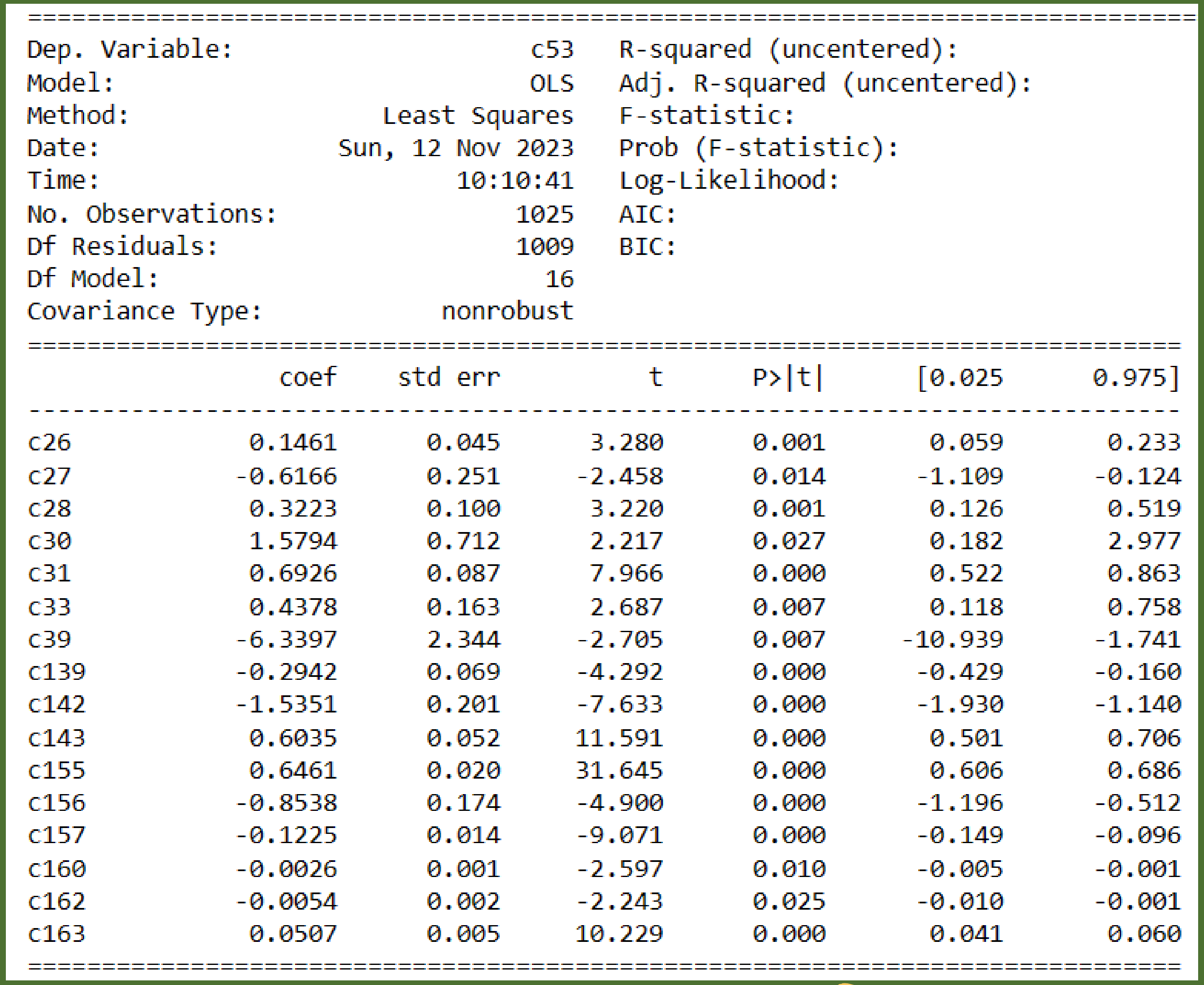


```
==============================================================================================
Dep. Variable:                       c52   R-squared (uncentered):
Model:                               OLS   Adj. R-squared (uncentered):
Method:                    Least Squares   F-statistic:
Date:                   Sun, 12 Nov 2023   Prob (F-statistic):
Time:                           10:10:41   Log-Likelihood:
No. Observations:                   1025   AIC:
Df Residuals:                       1008   BIC:
Df Model:                             17
Covariance Type:               nonrobust
==============================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
c26            0.5328      0.055      9.693      0.000       0.425       0.641
c27           -0.5169      0.251     -2.059      0.040      -1.010      -0.024
c28            0.1626      0.060      2.727      0.006       0.046       0.280
c29           -0.5361      0.056     -9.591      0.000      -0.646      -0.426
c30            1.9988      0.535      3.736      0.000       0.949       3.049
c31            0.3901      0.056      6.982      0.000       0.280       0.500
c32           -0.3686      0.132     -2.788      0.005      -0.628      -0.109
c33            0.8822      0.291      3.029      0.003       0.311       1.454
c39            6.1486      1.226      5.016      0.000       3.743       8.554
c139          -0.2603      0.037     -6.949      0.000      -0.334      -0.187
c142          -0.8165      0.125     -6.556      0.000      -1.061      -0.572
c143           0.2465      0.029      8.577      0.000       0.190       0.303
c155           0.0742      0.012      6.321      0.000       0.051       0.097
c157          -0.0655      0.007     -9.198      0.000      -0.079      -0.052
c158           0.0750      0.015      5.079      0.000       0.046       0.104
c160          -0.0029      0.001     -4.721      0.000      -0.004      -0.002
c161           0.0073      0.001      7.165      0.000       0.005       0.009
==============================================================================================
```

Summary Results!

Controlling and Monitoring c53 and c54 Columns

- The correlation heatmap reveals a high correlation (0.96) between 'c53' and 'c54,' allowing us to model them together. Using 'c53' data and Controllable Data as X, we iteratively drop parameters with P-values > 0.05, excluding c32, c158, c161, and c29. The resulting model achieves an R^2 value of 0.948.
- Similar to previous models, we identify crucial parameters by examining the absolute values of their coefficients
- Emphasizing c39, c30, c142, and c156, the model yields low MSE values (7.163 for c53, 9.862 for c54), indicating effective control of vibrations in both c53 and c54.



```
==============================================================================
Dep. Variable:                    c53   R-squared (uncentered):
Model:                            OLS   Adj. R-squared (uncentered):
Method:                 Least Squares   F-statistic:
Date:                Sun, 12 Nov 2023   Prob (F-statistic):
Time:                        10:10:41   Log-Likelihood:
No. Observations:                1025   AIC:
Df Residuals:                    1009   BIC:
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
c26            0.1461      0.045      3.280      0.001       0.059       0.233
c27           -0.6166      0.251     -2.458      0.014      -1.109      -0.124
c28            0.3223      0.100      3.220      0.001       0.126       0.519
c30            1.5794      0.712      2.217      0.027       0.182       2.977
c31            0.6926      0.087      7.966      0.000       0.522       0.863
c33            0.4378      0.163      2.687      0.007       0.118       0.758
c39           -6.3397      2.344     -2.705      0.007     -10.939      -1.741
c139          -0.2942      0.069     -4.292      0.000      -0.429      -0.160
c142          -1.5351      0.201     -7.633      0.000      -1.930      -1.140
c143           0.6035      0.052     11.591      0.000       0.501       0.706
c155           0.6461      0.020     31.645      0.000       0.606       0.686
c156          -0.8538      0.174     -4.900      0.000      -1.196      -0.512
c157          -0.1225      0.014     -9.071      0.000      -0.149      -0.096
c160          -0.0026      0.001     -2.597      0.010      -0.005      -0.001
c162          -0.0054      0.002     -2.243      0.025      -0.010      -0.001
c163           0.0507      0.005     10.229      0.000       0.041       0.060
==============================================================================
```

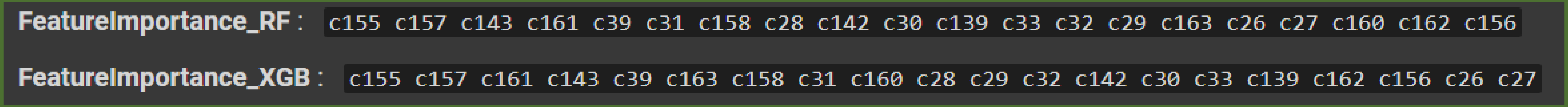
Summary Results!

Random Forest Regressor to Determine Important Features!

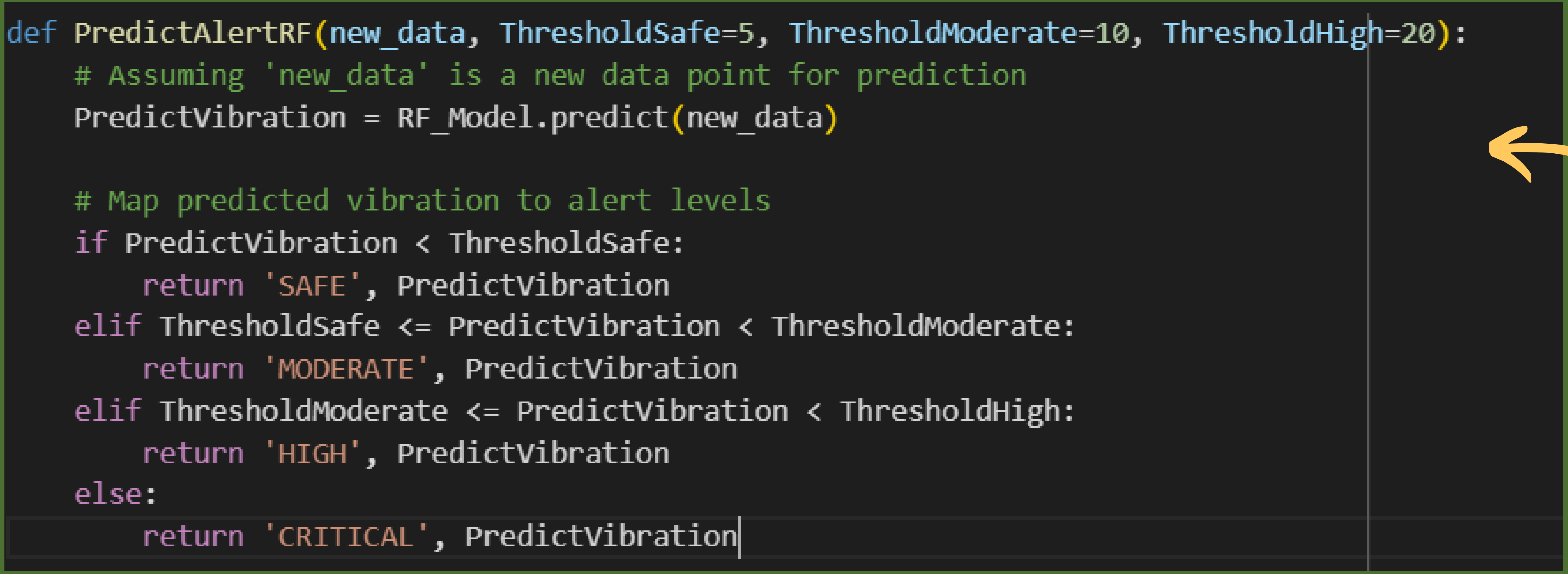
- Now we tried to make another model to predict and find out the most important features for all four vibrational classes as whole.
- Applying Random Forest Regressor on ControllableData, we identify the top 5 important features—c155, c157, c143, c161, and c39—using the `feature_importances` function.
- Feature importance is calculated based on how much the feature contributes to reducing the errors in the model. The feature importance scores are generally normalized to sum to 1
- These features play a vital role in both classification and control. We attain a impressive R2 value of 0.96 via this approach.

REPLICATING RESULTS WITH XGBRegressor!

- Another approach taken that provides the same results is the XGBRegressor of the xgboost library. This stands for Extreme Gradient Boosting which is an efficient and scalable implementation of gradient descent.
- Gradient boosting is an ensemble learning technique that combines the predictions of multiple weak learners (typically decision trees) to create a strong predictive model.
- This trained model gives us the exact same top 5 features namely c155 c157 c161 c143 c39 furthur validating our methods. While the complete lists of the feature importances differ slightly for both the appraoches, they align with each other fairly well giving us confidence in our findings.



```
FeatureImportance_RF :  c155 c157 c143 c161 c39 c31 c158 c28 c142 c30 c139 c33 c32 c29 c163 c26 c27 c160 c162 c156

FeatureImportance_XGB :  c155 c157 c161 c143 c39 c163 c158 c31 c160 c28 c29 c32 c142 c30 c33 c139 c162 c156 c26 c27
```



```python
def PredictAlertRF(new_data, ThresholdSafe=5, ThresholdModerate=10, ThresholdHigh=20):
    # Assuming 'new_data' is a new data point for prediction
    PredictVibration = RF_Model.predict(new_data)

    # Map predicted vibration to alert levels
    if PredictVibration < ThresholdSafe:
        return 'SAFE', PredictVibration
    elif ThresholdSafe <= PredictVibration < ThresholdModerate:
        return 'MODERATE', PredictVibration
    elif ThresholdModerate <= PredictVibration < ThresholdHigh:
        return 'HIGH', PredictVibration
    else:
        return 'CRITICAL', PredictVibration
```

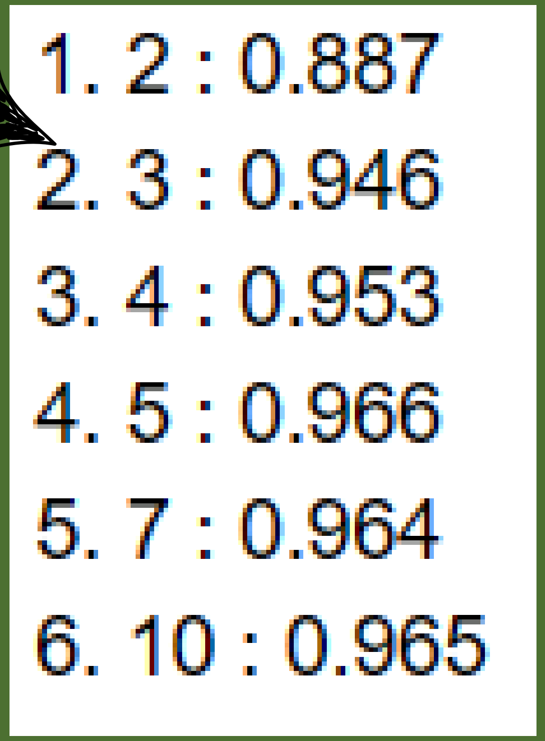
We created a Alert Function based on our above model to predict the Vibrations of the data we feed into this function

3. ML Model to Predict Specific Energy Level and to find out the most important features

Specific Energy Prediction and Modelling

- The parameter c241, measuring specific energy (energy consumed per unit output), is crucial. Utilizing RandomForestRegressor, we model specific energy with Controllable and Operating parameters.
- The `feature_importance_` function identifies key independent features for an efficient predictive model with minimal parameters.
- From the above table, we can see that the best R^2 value is observed for 5 independent variables. Thus for our improved and efficient model we choose the Top 5 parameters.

R2 value for different number of parameters

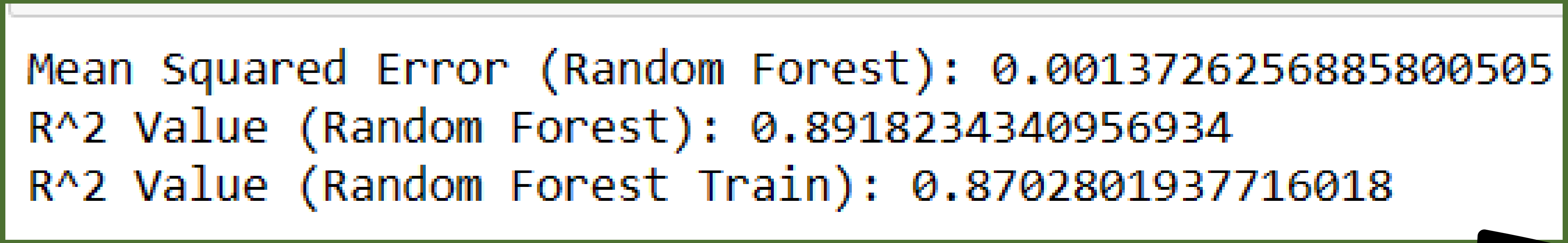


```
1. 2 : 0.887
2. 3 : 0.946
3. 4 : 0.953
4. 5 : 0.966
5. 7 : 0.964
6. 10 : 0.965
```

Top 5 parameters:
['c151', 'c65', 'c192', 'c117', 'c18']

FINAL RESULTS FOR SPECIFIC ENERGY PREDICTION

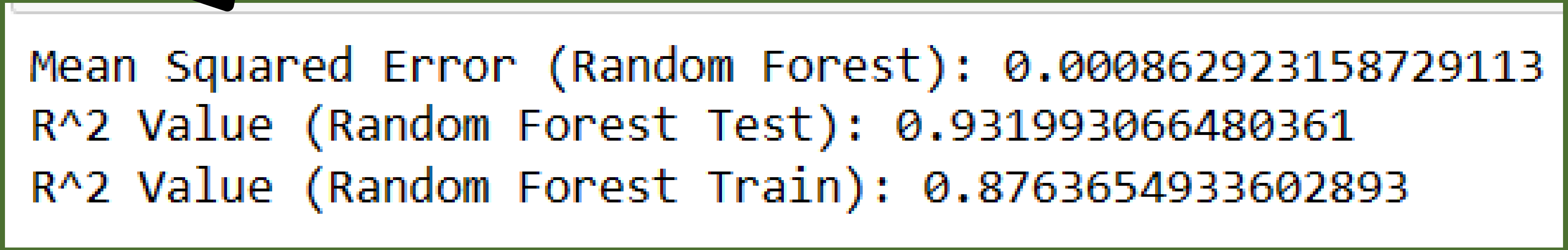
- Based on the described method in the previous slides we run our ML model and obtain the different regression statistics
- In order to emphasize the importance of feature selection one can compare the values of MSDE and R2



```
Mean Squared Error (Random Forest): 0.0013726256885800505
R^2 Value (Random Forest): 0.8918234340956934
R^2 Value (Random Forest Train): 0.8702801937716018
```

Reggression Stats Before Feature Selection

Reggression Stats after Feature Selection



```
Mean Squared Error (Random Forest): 0.000862923158729113
R^2 Value (Random Forest Test): 0.931993066480361
R^2 Value (Random Forest Train): 0.8763654933602893
```

THANK YOU