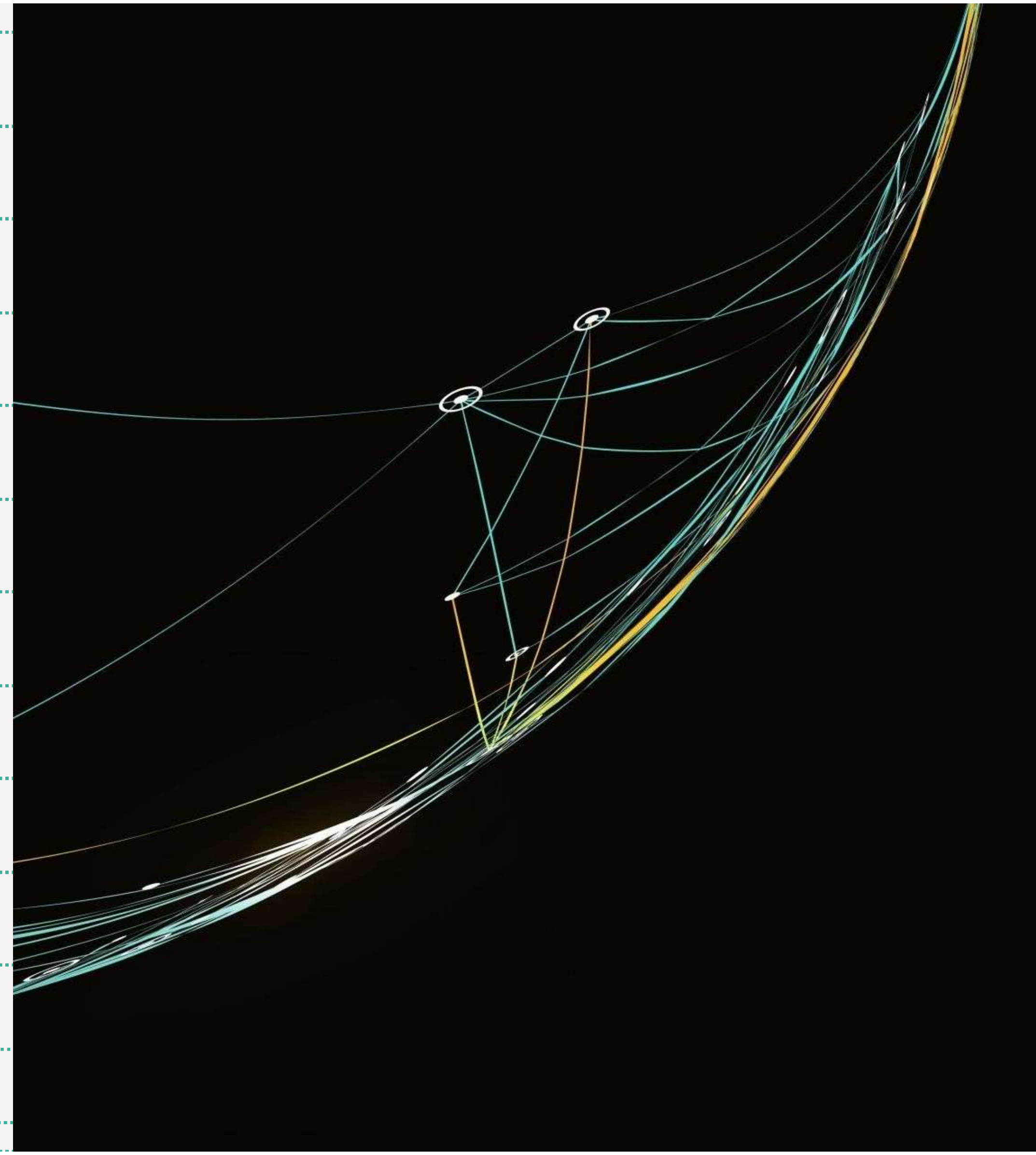
# DS203 E7v2

1. Group: 40
2. Rahul Jarupla(22B2180)
3. Madhava Sriram(22B1233)
4. Hardik Gohil Pratap(22B1293)

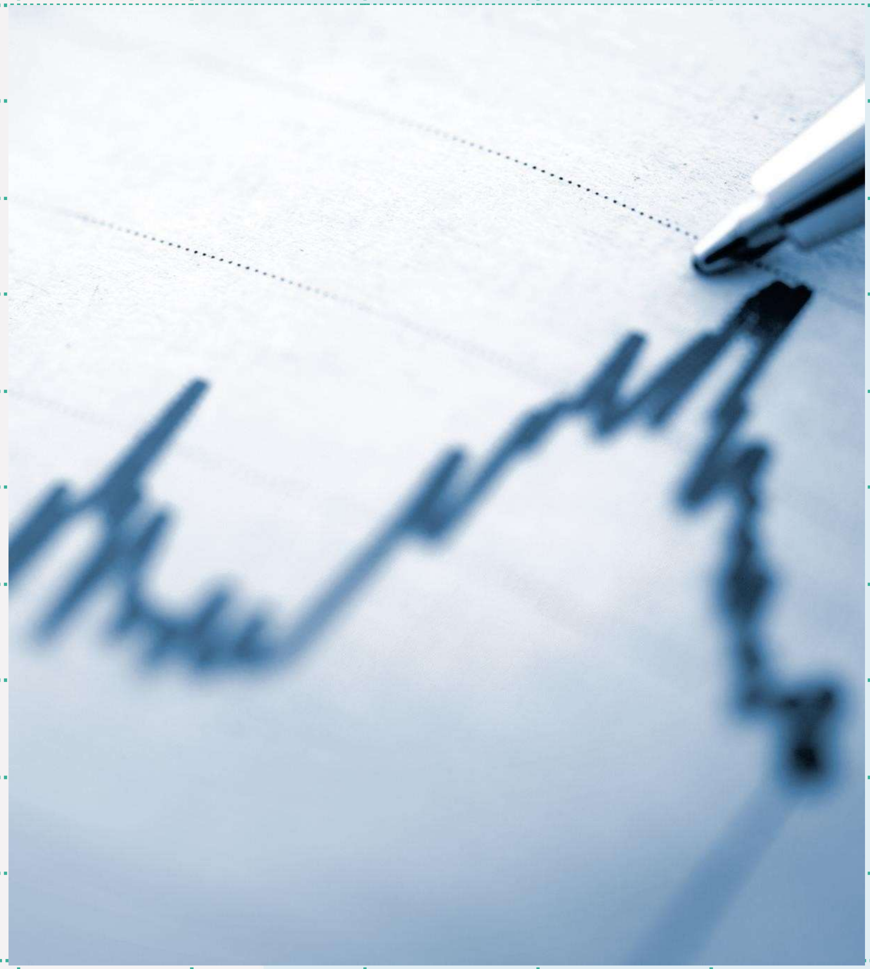


# Problem Description(Statement)

- 1)The dataset comprises real-time information derived from a transformer at a solar power generation site. The data is collected at regular intervals of 5 minutes, spanning a duration of 285 days.
- 2)The issue at hand pertains to the presence of missing and noisy data points within the dataset. Our approach involves employing data analysis tools to systematically identify, cleanse, and transform the compromised data into a more reliable state, utilizing the available high-quality data as a reference for refinement.

# Problem Description

1) Missing Data:
2) We were genuinely confused what to do with the missing data later we got an idea to either remove them or fill them we potentially addressed the identification of gaps in dataset
3) Inconsistencies in Bad and Good days:
4) We derived to a conclusion to classify on basis of noise values but there were some issues present we could have applied rolling mean/std deviation but for concise and preciseness we plotted graphs using rolling std deviation and classified a few by hand



# Summary of my solution

- 1)We first classified Days into

a) Really Good days –which need no corrections

b) Good days- They need a few corrections

c) Bad days- They have high noise in them and a lot missing values

We classified by noise values(Std deviation of second derivative of a date using graph we plotted)

2) We identified rolling std deviation also as a metric to judge good, bad days generally in the time 07:00 to 18:00 the rolling deviation was high for good days it was totally high for the bad days

3) We then predicted improved current using the metric above, then if the current for any day is missing we replace it by the improved current hence making good days really good and bad days better,

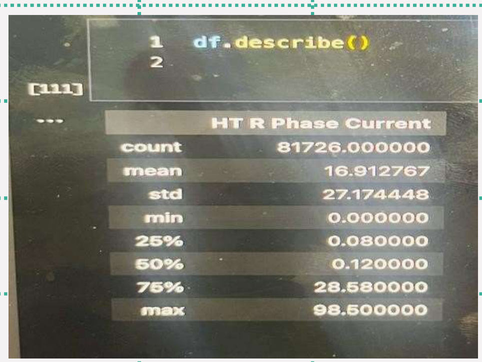
4) Making a Randomforestregressor and adapting it

# Exploratory Data Analysis

The data contained 82388 rows x 2 columns out of which 662 were NA's we have dropped them,

For better identification we divided the timestamp into date and time columns(time then being converted to hours and minutes)

We identified missing days as if there is no current for the whole day then it is a missing day we around had 38 missing days, for df.describe()



```
[111]  1  df.describe()
       2
```

| | HT R Phase Current |
|---|---|
| count | 81726.000000 |
| mean | 16.912767 |
| std | 27.174448 |
| min | 0.000000 |
| 25% | 0.080000 |
| 50% | 0.120000 |
| 75% | 28.580000 |
| max | 98.500000 |

# EDA- Identification of days

1)We found noise values for the whole dataframe excluding the missing days, and after many iterations and by looking at graphs we set a threshold

If 'Noise Value'<1,then it is 'Really Good'.

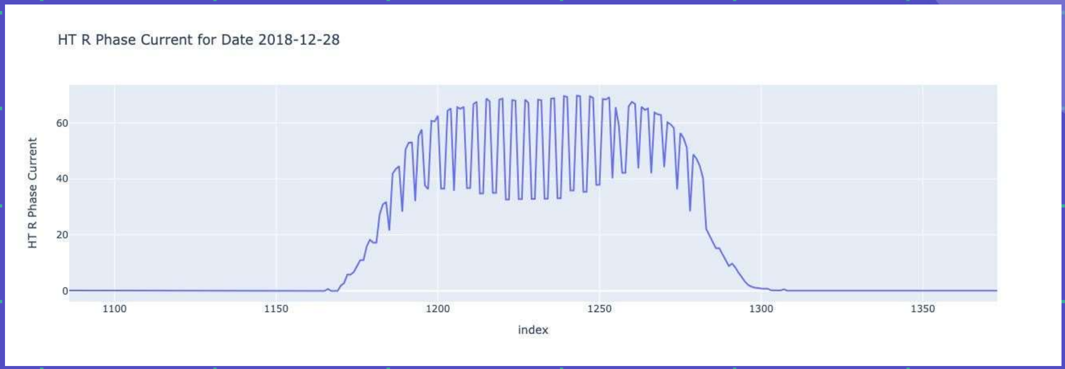
If 1<='Noise Value'<4.79 it is a 'Good Day'.

Else a 'Bad Day'.

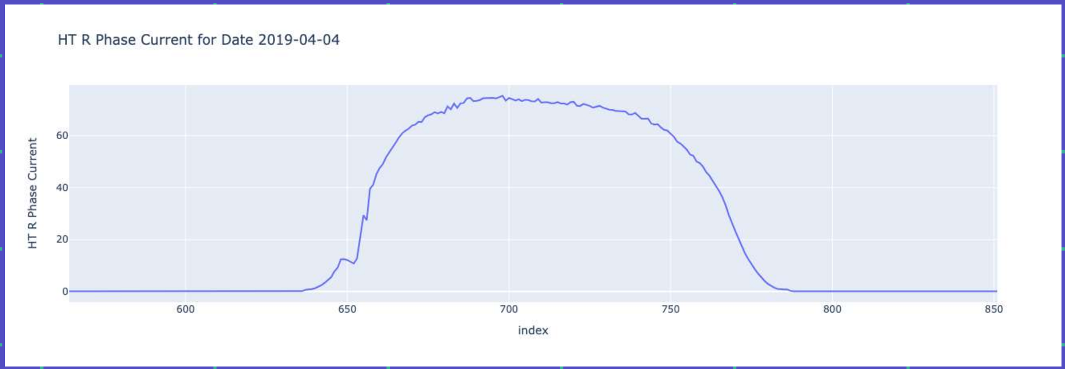
2) By same manual adjustments we found to have around 20 really good days , 42 good days and rest being bad days

3) As really good, good, bad days are identified we need to better them

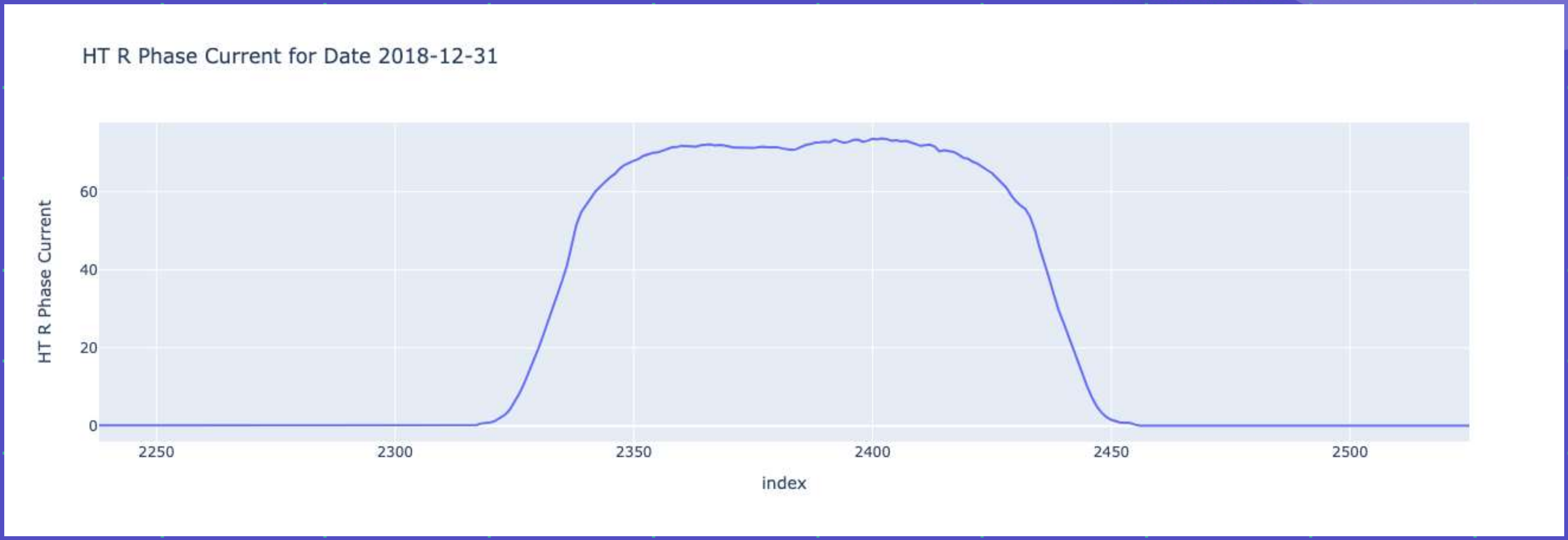
- 1)Bad day




- 2) Good day

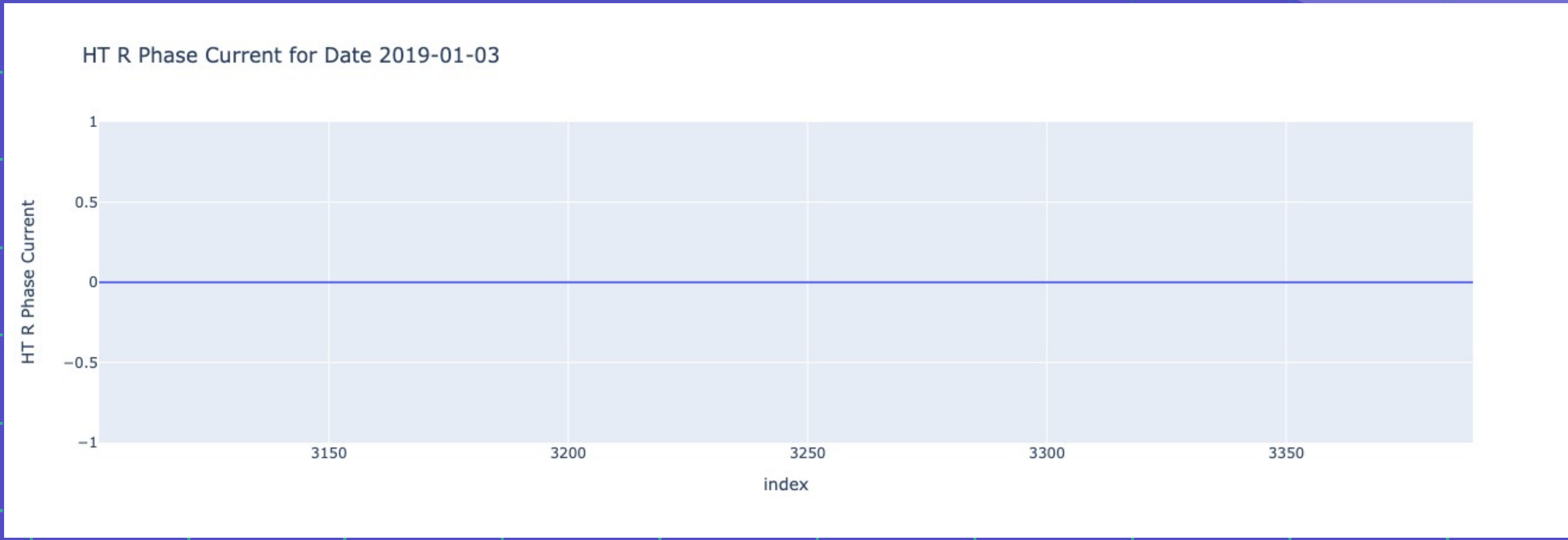


- 3)Really Good Day




The difference between 2 and 3 is their "noise" values, good day has noise value around 2.8 where as very good day has 0.6

- 4) Missing Day



# Data Preprocessing

- **1) Handling Missing Data**
- Utilized interpolation techniques to fill missing data points.
- Preserved temporal patterns to ensure continuity in the dataset.
- **Reasons:**
- Avoided data distortion by choosing interpolation over imputation.
- **2)Identification of Stable Features**
- Applied rolling standard deviation to assess feature stability.
- Features with consistently low standard deviation were considered stable.
- **Justification:**
- Stable features are indicative of data quality and reliability.
- **3) Fitted Average for Bad Days**
- Calculated a fitted average using data from good days.
- Applied the average to bad days to create a more stable representation.
- **Reasons:**
- Smoothed out fluctuations in bad days for improved consistency.
- Enhanced the overall quality of the dataset by making bad days more similar to good days.


# Why Rolling Std.dev?

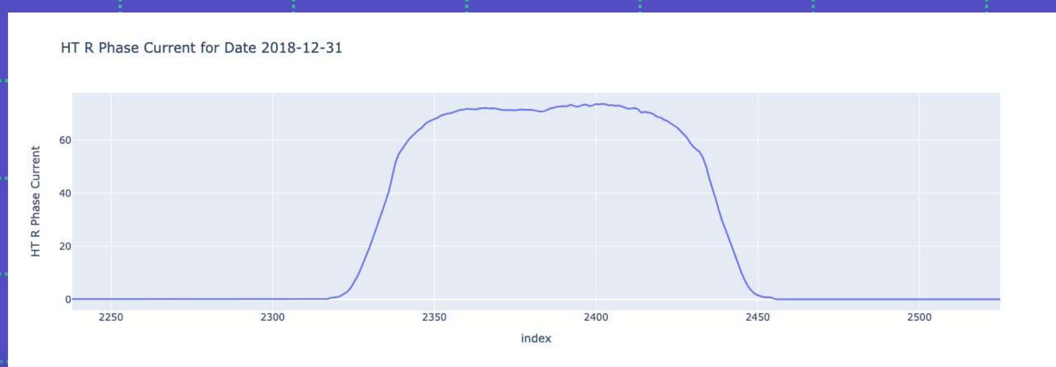
- Rolling standard deviation is sensitive to changes in data variability.
- Consistently low rolling standard deviation indicates periods of stability.
- It helps in identifying intervals where the data values are relatively constant and less prone to abrupt changes.
- It provides a measure of how much the data points deviate from their average within a specified time window.
- Rolling standard deviation is less affected by outliers compared to other metrics like the mean.

# Visual Representation

1) Bad Day with rolling std deviation

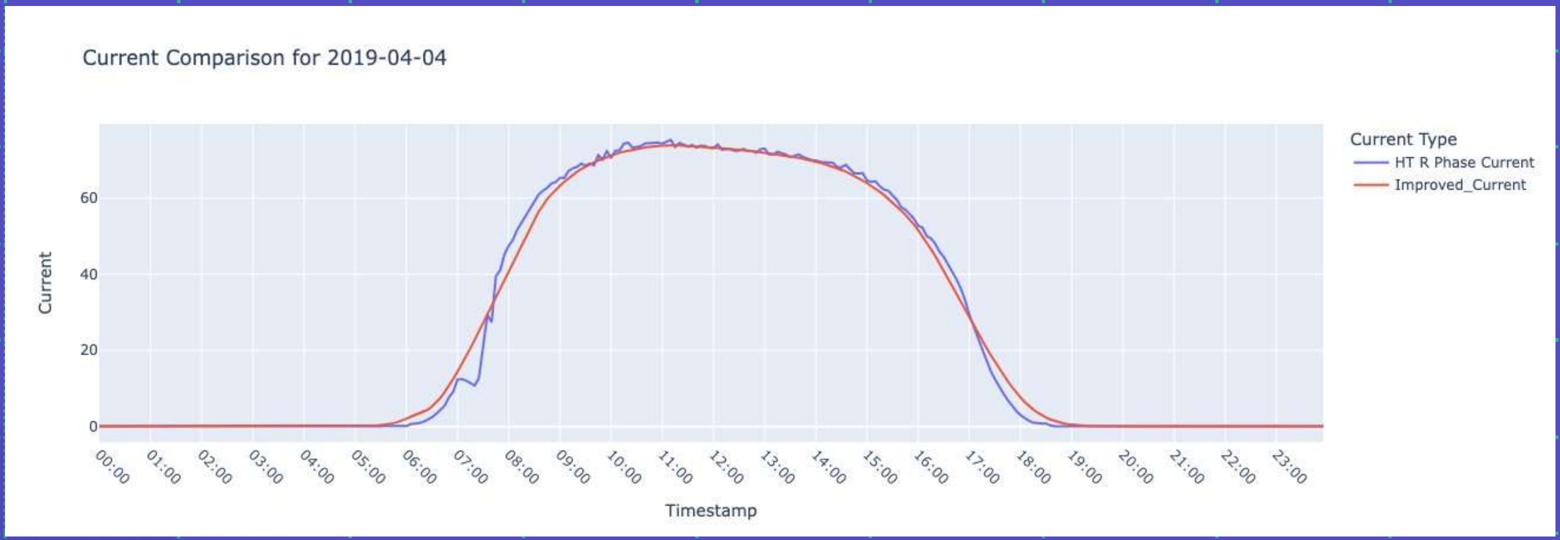
- 2) Good day with Rolling Deviation


- 3) Really Good Day with rolling std deviation

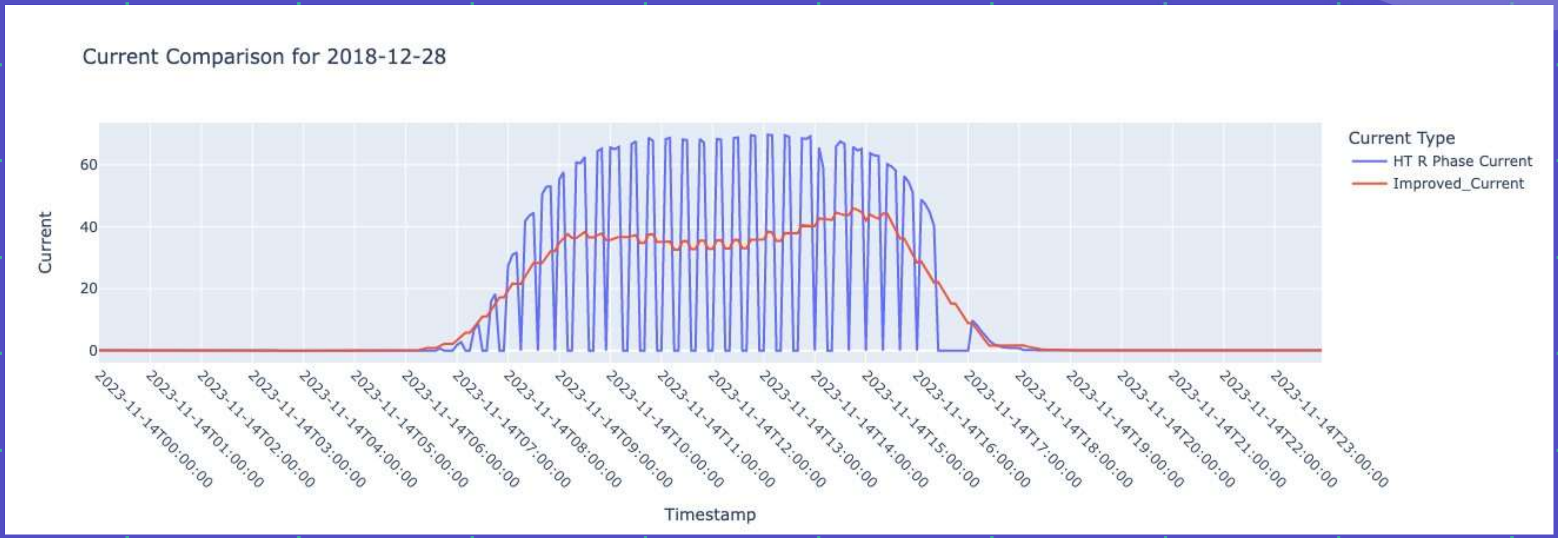


# Improved Graphs(After Replacing)

1) Good Day



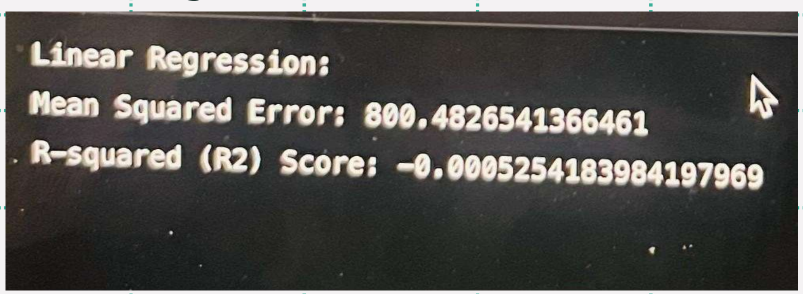
- 2) Improved Bad day



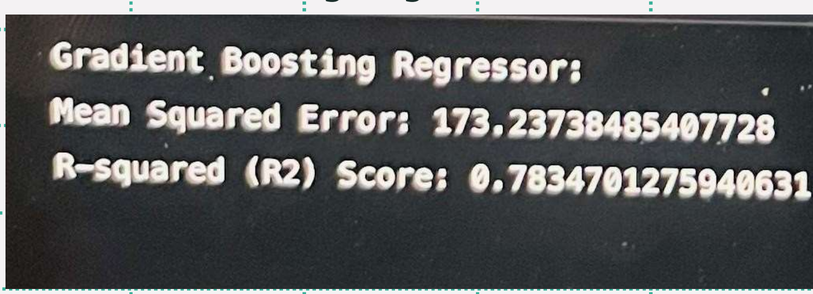
# How RandomForestRegressor?

- We chose R_squared, and MSE as 2 metrics inorder to evaluate how better is the model, we chose 3 regressors; Each of them is mentioned below

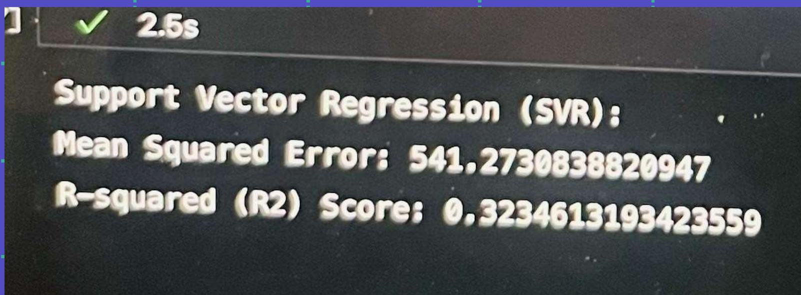
1) Linear Regression



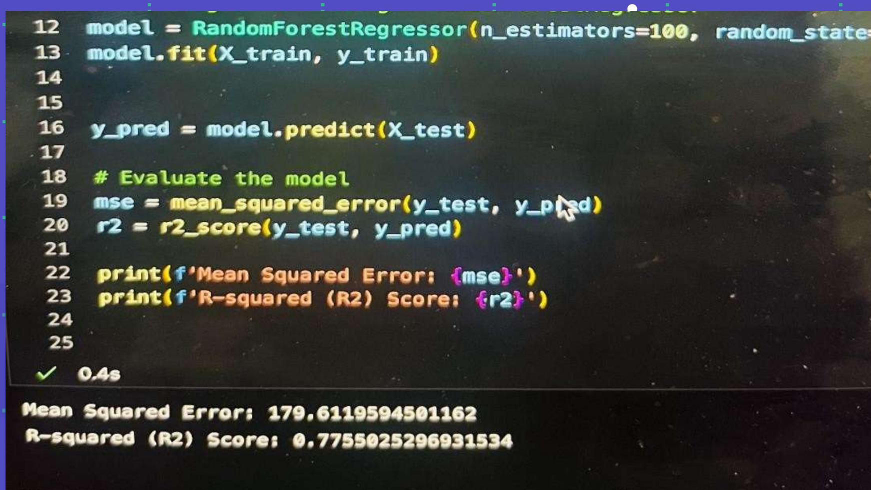

2) Gradient Boosting Regressor




- 3)Support Vector Regression




- 4) Random Forest Regression




Hence Randomforest regression is the best one

# Why RandomForestRegressor?

- RandomForestRegressor is capable of capturing non-linear relationships in the data preventing multicollinearity
- If it exists,RandomForestRegressor can handle multicollinearity well, where predictor variables may be correlated with each other.
- Use of RandomForest makes it less sensitive to outliers in the data.
- RandomForestRegressor provides a measure of feature importance, but in our case we used less independent variables, this feature is more helpful for more data
- RandomForest can handle imbalanced datasets without the need for resampling techniques.
- The most Important RandomForestRegressor tends to generalize well to new, unseen data.

# METRICS FOR MODEL

MSE for Accuracy Assessment:**

- \- Mean Squared Error (MSE) serves as a pivotal metric by measuring the average squared differences between predicted and actual values.
- \- Lower MSE values indicate a more accurate fit of the regression model to the dataset, offering a straightforward representation of its precision.

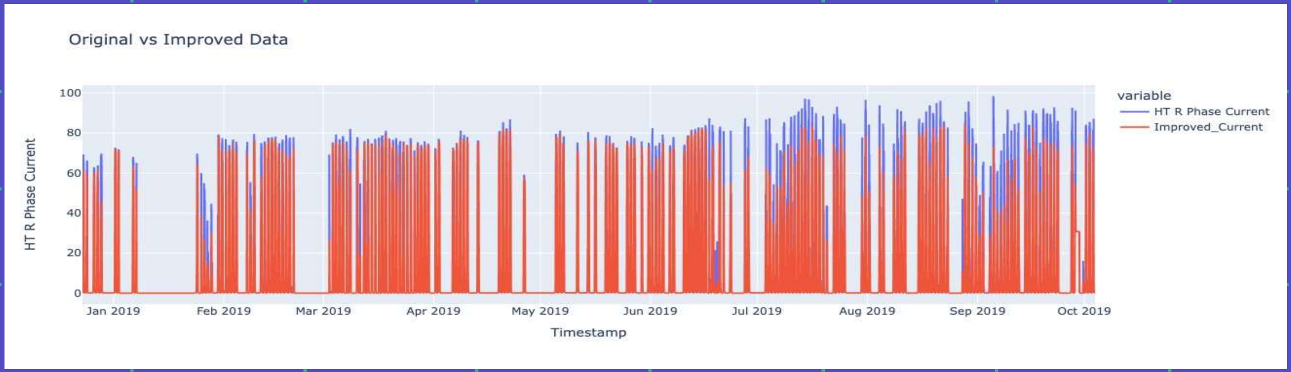
$R^2$ for Explanatory Power:**

- \- R-squared ($R^2$) gauges the proportion of variance in the dependent variable captured by the model, reflecting its explanatory power.
- \- A higher $R^2$ value signifies a stronger ability to explain the variability in the response variable based on the model's predictors.

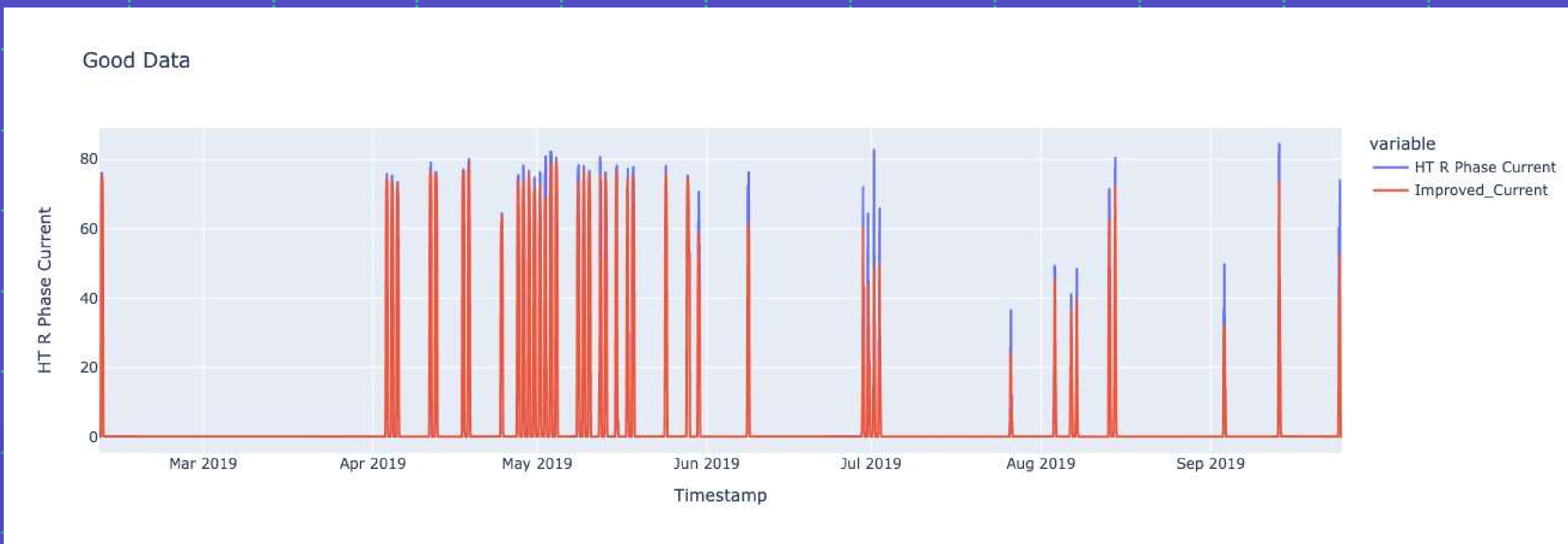
# METRICS FOR MODEL

- Comprehensive Insights:
- - The combination of MSE and $R^2$ provides a balanced and holistic evaluation of the regression model's performance.
- - MSE captures accuracy aspects, while $R^2$ sheds light on the model's explanatory capability, ensuring a thorough understanding of its overall effectiveness.
- - Precision Evaluation:
- - MSE offers a quantifiable measure of precision by assessing how closely predicted values align with actual observations.
- - $R^2$ complements this by offering insights into how well the model's predictors explain the variability in the response variable.
- Facets of Regression Analysis:
- - Utilizing both MSE and $R^2$ ensures a nuanced assessment of the regression model's performance, considering diverse facets such as predictive accuracy and explanatory strength.
- - This comprehensive approach enhances our understanding of the model's effectiveness in handling various aspects of regression analysis.

# Predicted and Improved Plots for Bad days




# Original v/s Improved Plots for Good days



# Best Approach?

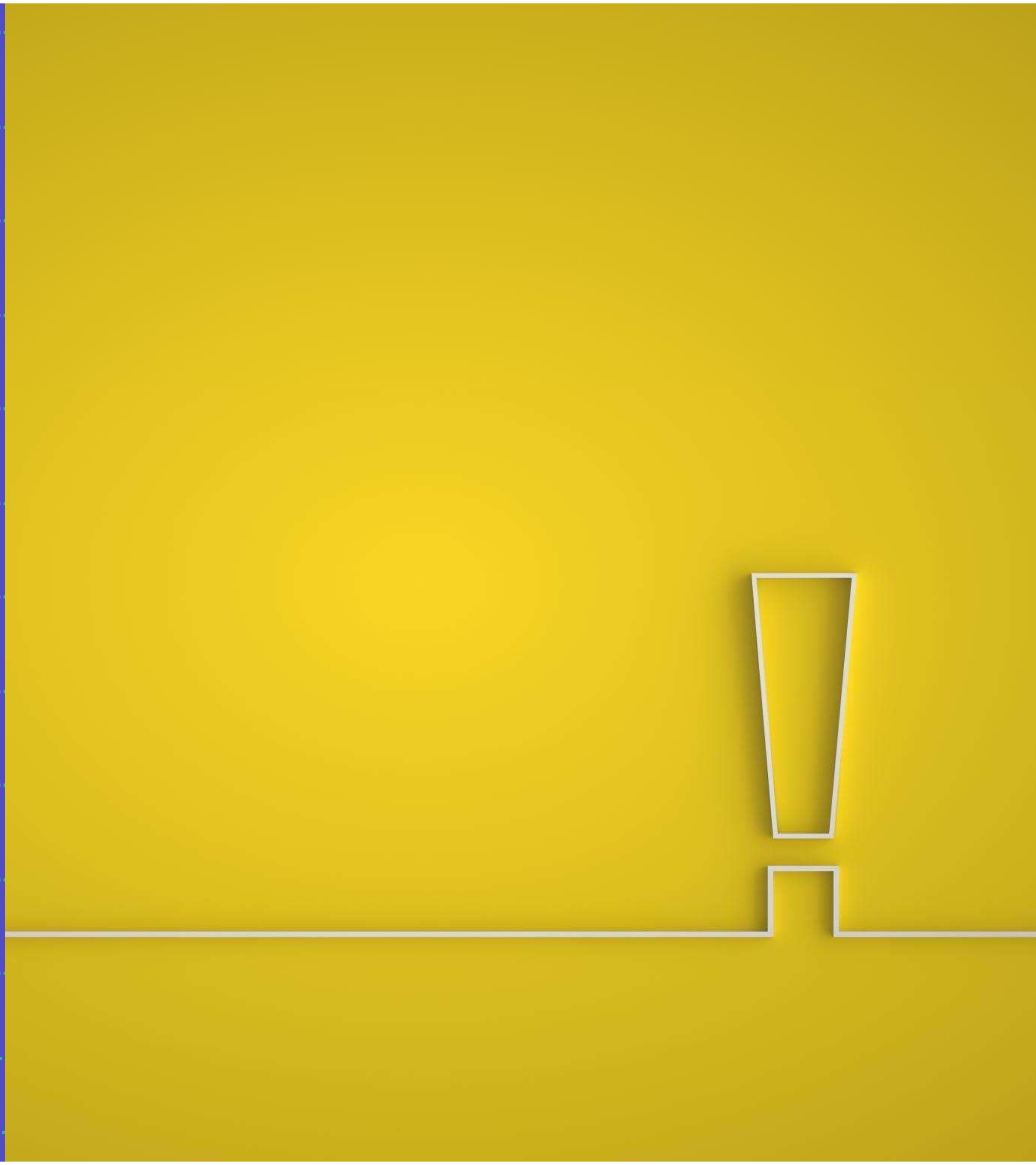
- My way is the best because it's super precise and not easily thrown off by weird data points, thanks to the model we picked.

- By combining these methods, we ended up with a dataset that's not just more complete but also way more solid and steady. Perfect for digging into analysis and modeling.

- I leaned more into math than just plain logic for my approach. It's a bit trickier than using regular algorithms, but it adds a layer of complexity that helps us handle things better.

# Alternate Approach

- Instead of using rolling std deviation, we could have replaced missing values with the median/mean value of whole dataset
- This could have been automated because here due to accuracy I did hand-picked few dates
- Instead of random forest regressor, **K-Nearest Neighbors (KNN) Imputation** I tried another way that uses a non-parametric method, focusing on local info. This can catch those small, tricky patterns that big-picture imputation models like RandomForestRegressor might miss. Picking between RandomForestRegressor and KNN imputation depends on what the dataset needs and what we're trying to achieve.
- Can also be done using Neural Networks it gets little complex but can get you similar results



# Challenges

We first found it complex to understand how to classify and all, then we decided we are going to classify by noise values

Then We faced the NA value error, we dropped the columns

Then the first time we did it, The model which worked well for good days and its prediction for good days failed for bad days It then took a while to process and due to some code mistake we finally did it!

We initially had a very big confusion regarding the "number" of good and bad days , then finally convinced us it depends on the criterion we took

# Challenges

- We then needed to choose independent and dependent variables, we first included Date using "ONE HOT ENCODER" in dependent variables but then dropped the idea due to lack of knowledge, the pro's and con's of it! , Then for dependent variable we took 'Improved Current' which worked for good days then as mentioned we changed it!
- We also thought to use Gaussian regression but got to know
- a)This model assumes the dependent variable to undergo gaussian function, I did never observe this kind for bad day's graph and also it is not sensitive to preventing outliers and it assumes a linear relationship between X and Y which I guess is not!!

# Should this improved dataset be used further?

- Yes, because It can beforehandedly do
  - Identification of stable features
  - Exceptional suitability for creating machine learningmodels
  - Bolstered stability for reliable model training
- It can improve sound to noise ratio
  - Reduction of noise in the dataset
  - Enhancement of the signal-to-noise ratio
- It can reduce variability
  - Resulting in reduced variability
  - Ensuring a consistent and robust dataset
- It can adapt quickly
  - Capturing accurate predictions for continuous target variables
  - Improving class separation in classification tasks
  - Forecasting future values based on historical patterns

# THANK YOU