Team ID- HA225398,
IIT (BHU),Varanasi.

Anjali Tiwari, Prachi Kumar, Raunak Pandey, Siddanth Shetty

Weldright-Techfest

INTRODUCTION

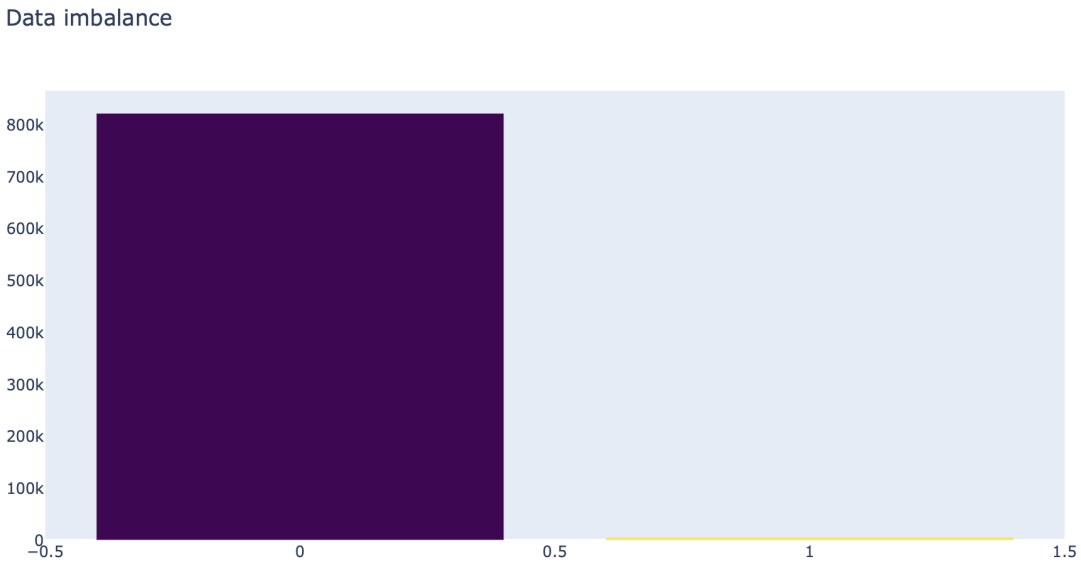
In this project, we are building algorithms using the parameters to predict materials' welding defects by applying ML models. The main aim is to help the Godrej Aerospace Team produce defect-free, high-precision spacecraft components. In the following content of the abstract, we present the methodologies used on the provided dataset with 827534 rows and 88 columns to build a helpful algorithm.

PRIMARY ANALYSIS OF THE DATASET

The first step of the model creation was the cleaning and preprocessing of the dataset. After performing various techniques, a clean dataset was obtained from which the following inferences were made:

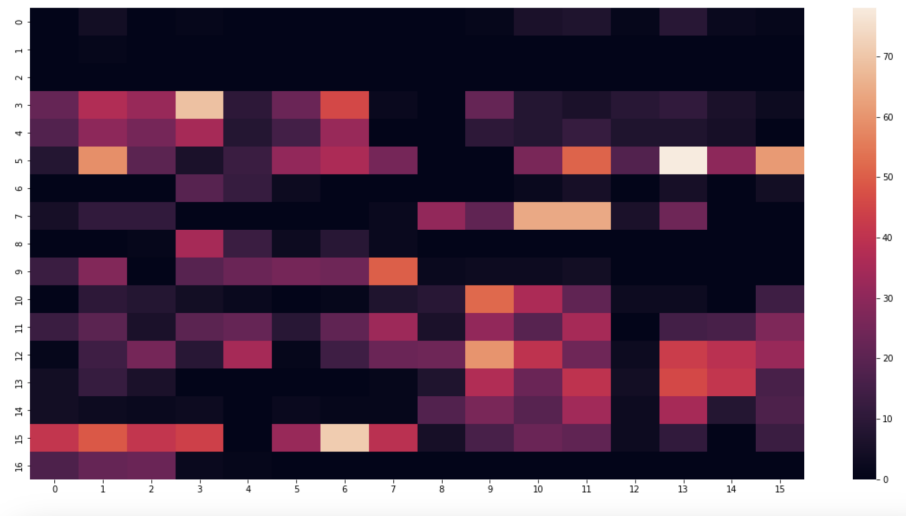
1. In the dataset, columns 13 to 87 only contain NaN values, and the first row of the data only describes the individual columns. Thus, columns 13 to 87 and row 1 can be dropped from the data frame for future analysis.
2. The dataset contains data for 24 days between 22/08/2019 and 19/09/2019, and 8 different welders are operating as per the given data.

3. A look at the defects column of the dataset tells us that there are primarily 2 kinds of defects observed in the welds-' Tungsten defect' and 'Porosity defect.'



(fig.1)

4. The dataset is heavily imbalanced.(refer fig.1)
5. A total of 4 operations and 16 different projects were undertaken during the period.
6. The data includes many welds that occur at nearly 0 Voltage.
7. Most of the defects occur in the month of September which according to our hypothesis, corresponds to the gradual wear and tear of machines.
8. On analysis of the defects on the basis of time, we found that **most of the defects occurred during the time period of 4 PM to 7 PM(refer to the heatmap below)**. This might be due to a number of reasons such as non optimum conditions for welding, machine fatigue during the time or poor employee performance during the time due to the passive attitude of the workers post-lunch. The company can look into it and dig deeper to find out the actual reason and in turn greatly cut down on the losses due to welding in these time slots.



Heatmap showing the intensity of defects during working hours(x-axis) and days (y-axis)

OUR APPROACH

Data Preprocessing

Firstly, we carried out an exploratory analysis of the dataset to find the factors that are more likely to affect the weld quality amongst those given in the dataset. Columns of Production and Machine were dropped as they are the same throughout the sample. The data was cleaned and preprocessed.

Columns containing corrupted values and values that seemed to provide no additional information to the model were removed.

Since we only need to predict the defects in welding accurately, we encoded the data in the defects column using LabelEncoder with 0 and 1 labels corresponding to no defect and a defect found in the welding process, respectively. We have now reduced the given problem to a binary classification problem.

Feature Engineering

The next step is the **Creation of Features** , that is, feature engineering.
A total of four different features were created:

1.porosityF:

$$porosityF = Flow * Job\,Temp * Humidity$$

Porosity is linked to **gas flow, job temperature and humidity** according to the research papers, so creating a feature porosityF increases the likelihood of detection of porosity defect.

2.HeatInput: According to the standard formula, heat input is calculated as the product of Voltage and Current divided by the Speed of welding.

$$Heat\ Input(here) = \frac{Voltage * Current}{Flow}$$

However, since speed is unknown, a modified version of the formula is used where flow has replaced speed

3.Power: It can be calculated as the absolute value of product of Voltage and Current.

$$Power = absolute(Voltage * Current)$$

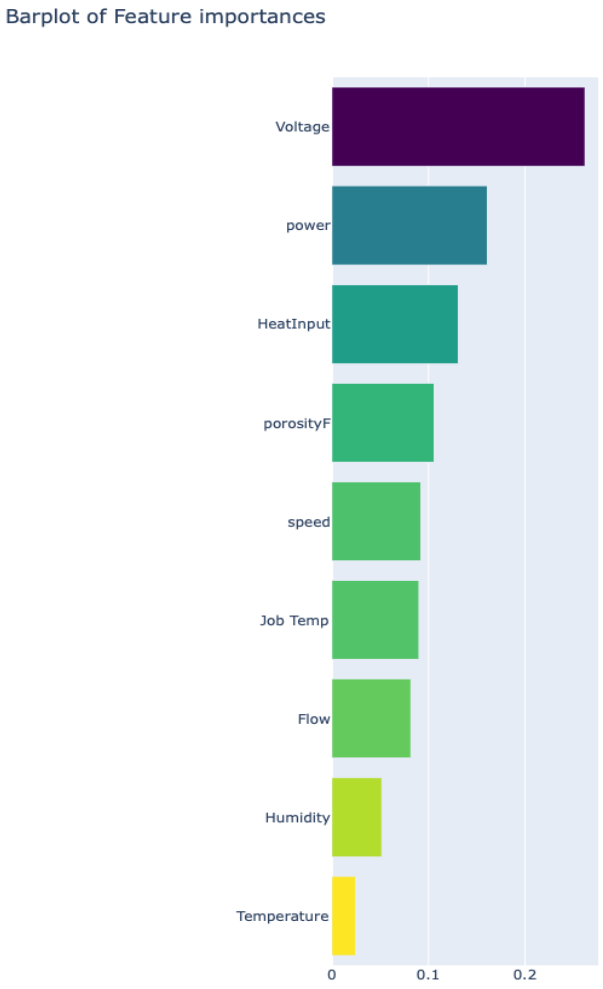
4. (weld) Speed:

$$(Weld)Speed = \frac{Distance(assumed\ Constant)}{Time\ Difference}$$

Technically defined as the speed with which the welder performs the welds, here information about the length of each weld was not given so it was assumed that each weld length is the same.

Label Encoding is done when categorical features need to be fed to the model; here it was done for Employee Code, Operation Number and Production Number.
However it led to poorer performance of the model and hence was dropped in the end.

Feature Selection



Features were selected based on the relative importance.

RELEVANT TECHNIQUES AND PERFORMANCE COMPARISON

Balancing the Classes

The problem statement is an example of rare class detection, so methods of **oversampling** were used to increase the minority class population density.
SMOTE(Synthetic Minority Oversampling Technique) was applied to improve the class imbalance.

Making Predictions

Based on the nature of the data, we tried training it using various classifier models such as SGDClassifier, RandomForestClassifier, CatBoost, XGBclassifier, Artificial Neural Networks,Angle-based Outlier Detection, and K Nearest Neighbours.

Amongst them **Angle-based Outlier Detection** and **Random Forest Classifier** seemed to work well for the model providing the most accurate results while making predictions.

ABOD(Angle-base Outlier Detection)

The model's f1 score satisfies the criteria of being >0.8.

The weighted average f1 score of the model is pretty high and unrealistic as the data was highly imbalanced and the entries for no defects are way greater than those with defects, hence macro f1 score is also calculated as we need to treat both the classes equal , but since weighted average f1 score has been mentioned as a judging criterion , so it's necessary to calculate.

```
In [106]: f1_score(y_test, pred, average='weighted')
Out[106]: 0.989089980985016

In [107]: f1_score(y_test, pred, average='micro')
Out[107]: 0.9927177832438485

In [108]: f1_score(y_test, pred, average='macro')
Out[108]: 0.4981727927513406
```

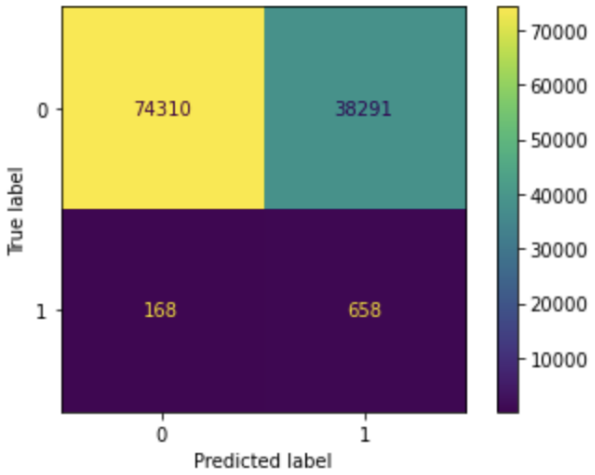
However the analysis of it's confusion matrix[1] would tell us that the model performs extremely poorly when it comes to actual detection of defects.Hence we suggest Random Forest Classifier.

[1] see the MODEL.ipynb

Random Forest Classifier

This model had a weighted F1 score of 0.78-0.85[2], which satisfies the criteria.

However it does an excellent job of predicting the defects.



As we can see the model has successfully predicted **658 defects** out of 826 defects in total. This gives it an accuracy of **79.66%** .

```
In [106]: from sklearn.metrics import f1_score
          f1_score(y_test, pred, average='weighted')
Out[106]: 0.7897426326151074

In [107]: f1_score(y_test, pred, average='micro')
Out[107]: 0.6621703827131107

In [108]: f1_score(y_test, pred, average='macro')
Out[108]: 0.4146831036218226
```

Here are the different F1 scores for the model. Hence the model is extremely successful in finding out the defects.

[2] The F1 score is not constant due to stochastic nature of the algorithm

ROI(Return On Investment)

Breakdown Of Costs

In a properly functioning welding operation, costs are typically broken down like this:
Labour: 85 %
Filler metals: 6 %
Raw materials: 4 %
Shielding gas: 3 %
Electric power: 2 %

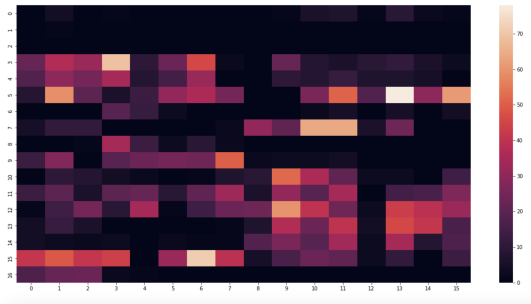
When a missed or defective weld goes undetected, these costs escalate at every stage in the welding operation and beyond.**However the monetary damage won't be limited to a single weld; suppose a defective weld is catastrophic and leads to failure of the aircraft or defence vehicle. In such a case the cost would easily amount to hundreds of millions if not billions of dollars. Lawsuits and alike would also follow.**

The company can see the following benefits if it chooses to go forward with our suggestions:

1. Labour:According to our calculations Employee Number 382617 is the least efficient; so much so that his defective work alone is more than that of the top three performers combined.

One replacing him with worker who is as efficient as average of top three performers the the company is expected to save 300-400 defects[3]Thus reducing the chance of catastrophic failure.

2. We found that **most defects occur in the time period of 4PM to 7PM.** The company can use this insight to reduce defects during this time and this can save upto 60 defects[4] over a day.



Heatmap showing the intensity of defects during working hours(x-axis) and days (y-axis)

[3] With 0.98 efficiency ;over a month where an employee welded 40,000 times on average he would contribute to **440** defects alone compared to the **top three performers** who combined would produce **400** defects only.Alone the performer with average of top 3 would produce 100-150 defects.
[4] Refer to Feature Extraction & Business Analytics.ipynb

So summing up after these insights the company saves 60 defects a day and 300-400 defects on the basis of employees.
So over a month (60*30+300=2100) that is more than 50% of defects can be reduced.

Calculations

$$Amount\ Invested = Cost\ of\ Raw\ material + Cost\ of\ Power\ Consumed + Labour\ Cost + Other\ operational\ Cost$$

And ROI(Return Of Investment is:

$$ROI = \frac{Net\ Income}{Cost\ of\ Investment} * 100$$

Now, the total amount invested is used for welding with as well as without defects
Breaking down these:
1.**Cost of Power Consumed**-

```
In [163]: trainn['Power'].sum()
Out[163]: 34046045.22554

In [164]: trainy['Power'].sum()
Out[164]: 578142.35524
```

Thus, **1.669%** of power is consumed in production of defective pieces, thus an equal percentage of power cost.Similarly, labour cost and raw materials are wasted.Now, since this model predicts 60 percent of defects , the return on investment in this model is going to be decent over a large time frame and large factory data.
Hence,

*ROI=(income from good welds-(investment in good weld + investment in defective welds))/(investment in good weld+investment in defective weld)*100*

The expenditure in welding defects is going to decrease after deploying this model and hence the roi of the company is definitely gonna increase.

TCO(Total Cost of Ownership)

Operational Costs of a Machine Learning Solution				
	Bare-bones		MLOps Framework	
Model Infrastructure	*A single machine in the cloud with no load management*	***$9,000 / yr***	*Redundant machines with a load balancer, or Kubernetes-type cluster*	***$8,000 / yr***
Data Support	*Timed script executed on infrastructure to pull data*	***$6,750 (labor)***	*Independent data pipeline manager for continuous updates of analytic data*	***$10,000 (labor) + $3,200 / yr***
Engineering / Deployment	*Model copied from data scientists machine to cloud machine*	***$9,000 (labor)***	*Continuous integration and continuous deployment (CI/CD) system to pull model from registry*	***$24,500 (labor) + $516 / yr***
Total Investment	$15,750 (labor) + $9,000 / yr		$34,500 (labor) + $12,000 / yr	
5 year TCO	**$60,750**		**$94,500**	

The Total Cost of Ownership (TCO) is often **the financial metric that you use to estimate and compare ML costs**.It is divided into 3 parts, Acquisition Cost, Ownership cost and post-Ownership cost. Since this is an ML model , it does not need a large group of people at every factory to look after, a single team of Data scientists can look after every branch's data , so the management cost is pretty low. On top of that the AWS charges and acquiring charges are also gonna be low in comparison to the profits it will yield by predicting the defects.
The model was created using the sci-kit-learn library, which is open-source and free of cost. The rest of the prices are estimated using AWS infrastructure and third-party engineering for deployment support. Choosing the MLOps framework, the total cost would be $94,500.

Conclusion

The model is able to predict defects about **60%** of the time even before they have occured.

The f1 score of the model is 0.82[5] . The model when applied to practical uses will help in significant **reduction of errors** and help save companies resources.

Besides that our model provides key insights into the nature of defects as well. Once these suggestions are implemented the company can **reduce 50-60% of the defects**.Other than that the model can predict 60% of the defects so in total we will be able to bring down defective welds significantly.

[5] On averaging the F1 scores of Random Forest Classifier