TECHFEST ABSTRACT WELDRIGHT:

Group Member:

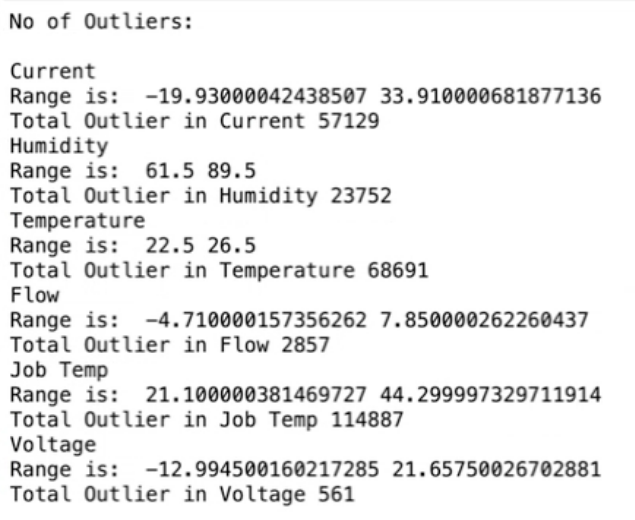
Pushkar Sawant

Sam Selvaraj (Team Leader)

Firstly, we observe that the dataset has many outliers. We find the Interquartile range to find the upper and lower bounds for each parameter. We then remove those values since our dataset is very huge and removing these values will not affect our model much. Also, some errors such as very high values for room temperature can be safely removed. We are removing humidity values above 100 assuming that the air is not super-saturated. We believe it is a fair assumption since super-saturation is possible only in conditions where there are no foreign particles for condensation.

Upper bound = Upper quartile + 1.5\*IQR

Lower bound = Lower quartile - 1.5\*IQR

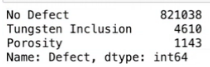


Secondly, we also see the dataset is unbalanced in its representation. We can use many different methods to tackle this. This is really important for training the ML model as we don’t want the model to output biased results. There are three main methods of doing this.

1. Over-sampling: In this method, we basically over-sample the minority categorical dependent variable. We duplicate those values so that we get a larger representation of our minority category. This increases our dataset size which is the last thing we would want to do.
2. Undersampling: In this method, we take a part of a dataset from the majority category so that we get an equal representation of all the categories. This method is effective and can be used by *ensemble techniques(*Random Forest, XGBoost, etc).
3. SMOTE(**Synthetic Minority Oversampling Technique):** This creates synthetic data of the minority category to populate it. It uses mean, median, mode, etc of the dataset to give random synthetic data. This increases the size too.

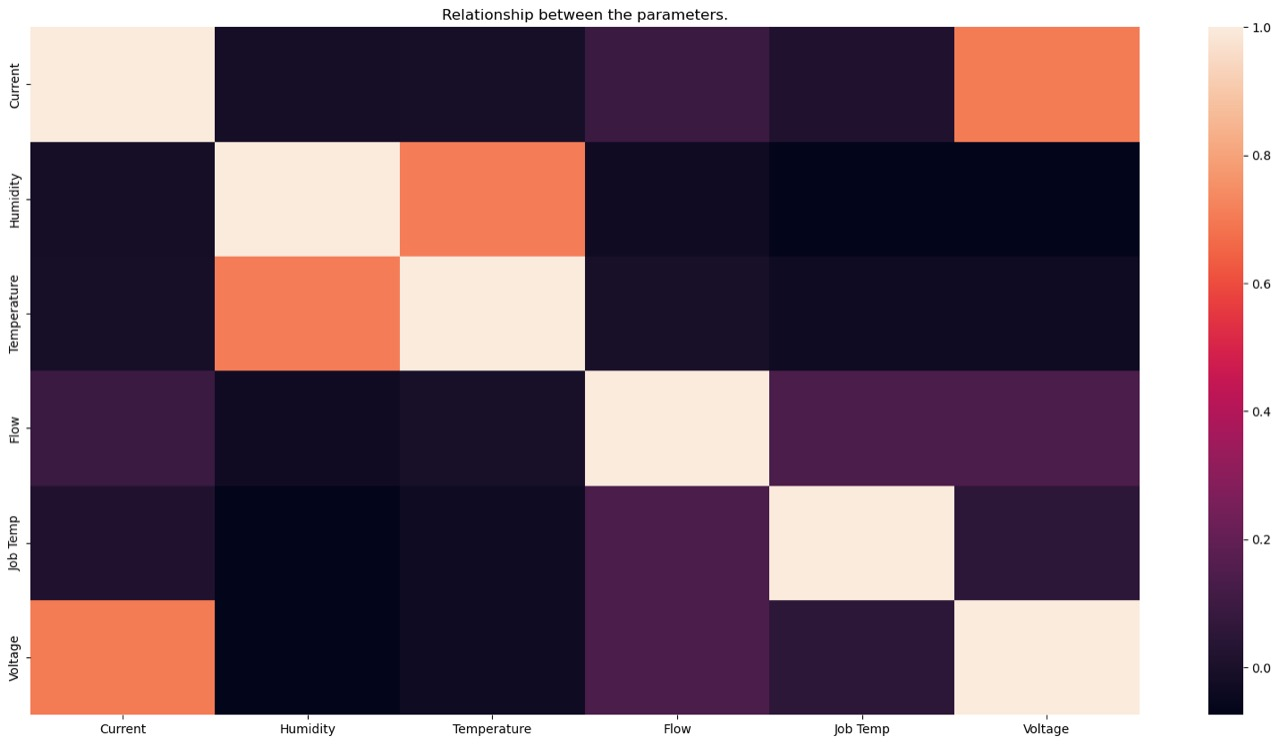
***From the above methods, we came up with a hybrid solution that is :***

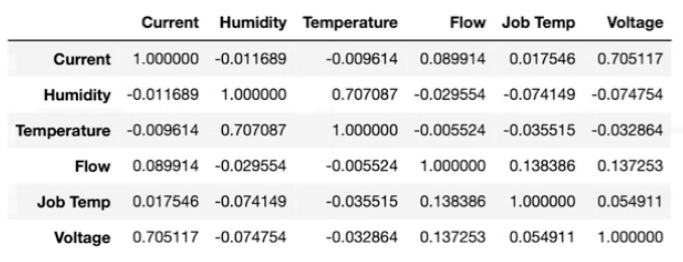
***“Undersampling the majority class and oversampling the minority class using SMOTE Technique would be the best method to train the model. Hence, the chances of biased output is reduced and would prove useful for creating a robust model.***



***OUR FINDINGS:***

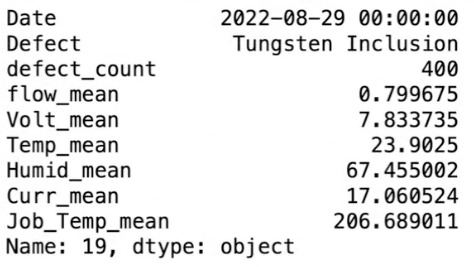
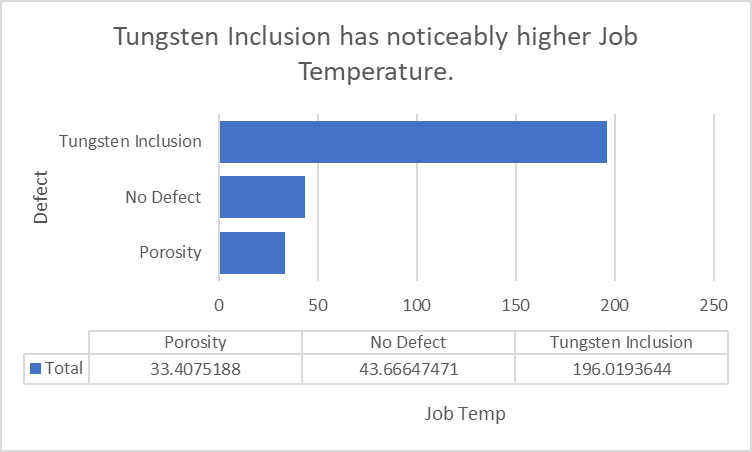
* Firstly, we made a heatmap of Pearson’s correlation of all the different independent variables. We found the correlation between Current and Voltage and also between Temperature and Humidity to be significantly higher than the other parameters(r-value). The correlation between current and voltage is very obvious due to Ohm’s law(V=IR). The positive correlation between ambient temperature and humidity is also a physical phenomena which can be confirmed from our dataset. Hence, no problem of multicollinearity.





* ***Inference regarding Job Temperature***

Tungsten Inclusion generally takes place when the welding job temperature is really high which causes the tungsten to melt and may enter the weld metal leading to defective welding. This physical property of welding is apparent in the below representation. We can see that Tungsten Inclusion has a noticeably higher ‘Job Temp’. ***According to this, we can suggest that a temperature range between 30-60 degree celsius would be ideal for welding. Highest defect count on the given data(29.08.22) has relatively high mean job temperature***

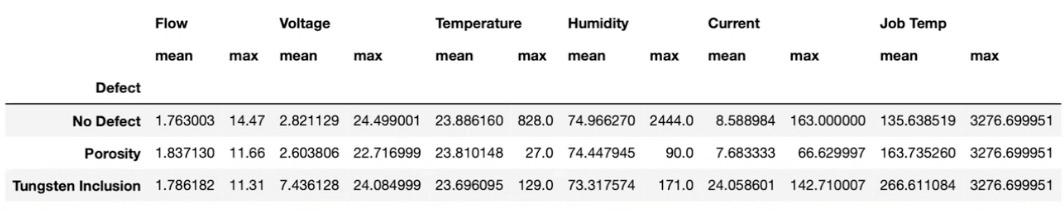


* ***Inference regarding Current and Voltage***

From the below table, we can also infer that Tungsten Inclusion has a higher mean Current compared to No defect. This, according to us, is because heat generated is directly proportional to the square of Current. Higher the current, higher the heat produced(high job temperature) and hence, higher chances of Tungsten Inclusion. Similarly, we also observe that average voltage is higher compared to no defect. This suggests that voltage might be affecting Tungsten Inclusion.

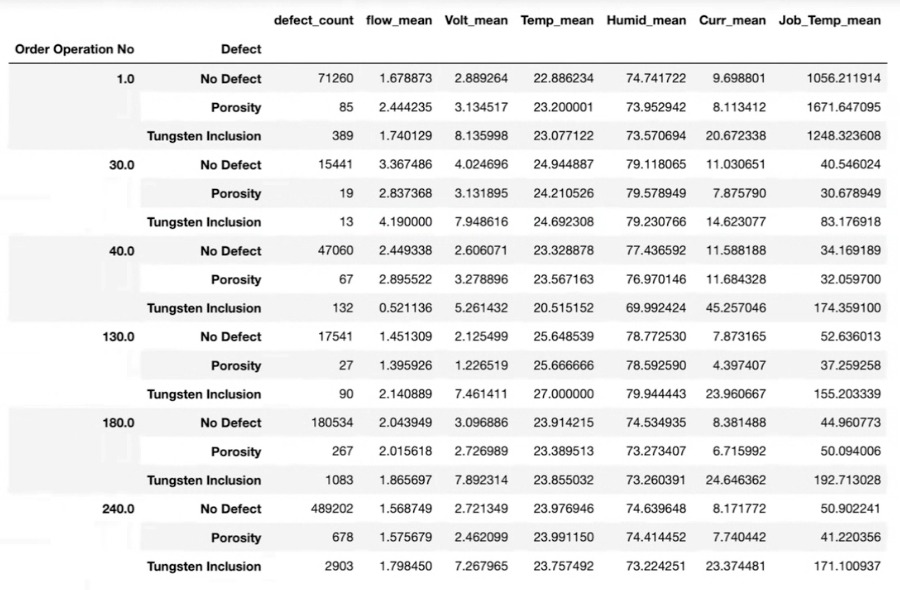
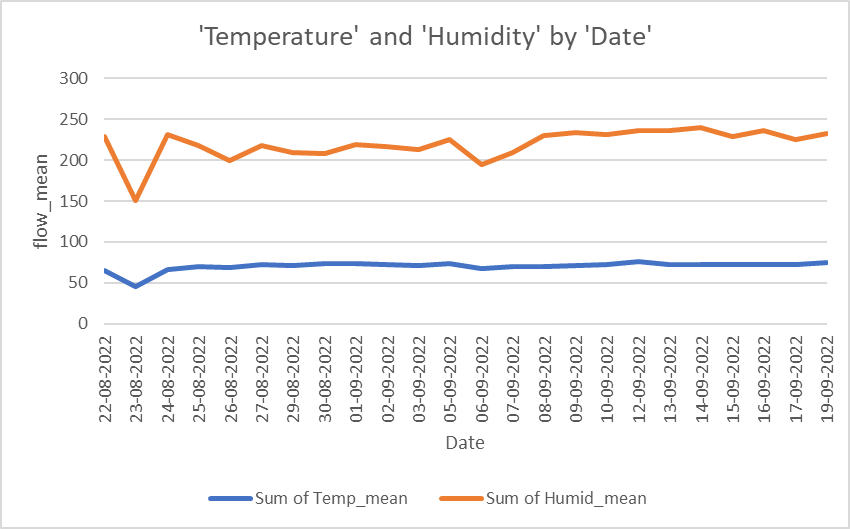
H = I2 RT

Where, I = Current , R=Resistance , T=Time and H=Heat energy

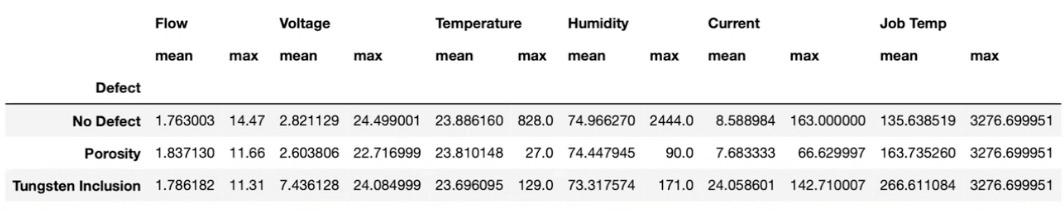


***Apart from this, we also observed few details in our data:***

1. The average ***temperature, humidity and flow rate*** is similar between No defect and Porosity throughout ***order operation numbers***. This suggests that temperature and humidity have minimal role in porosity. Ideally, high humidity is a major factor for porosity. But, in our dataset, we see that the range of humidity is very limited.

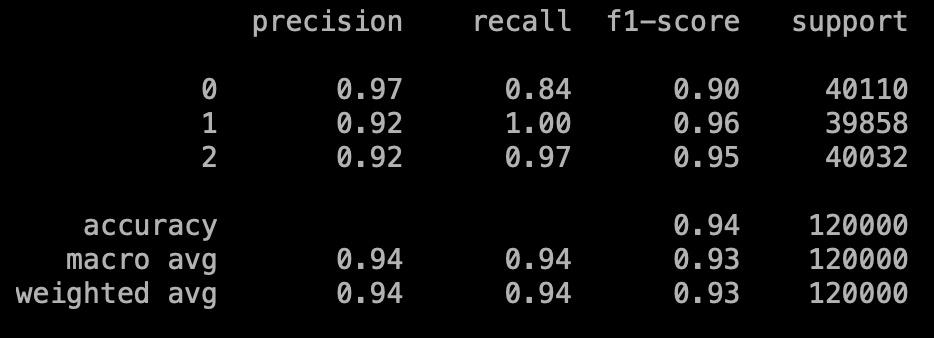
 

1. The average flow rate for porosity is higher than no defect and tungsten inclusion across the entire dataset.





*We decided to use the XGBClassifier model, this is because the model is extremely powerful when it comes to really huge datasets ,provides high accuracy and also helps in reducing the overfitting problem. With an accuracy of 94%.*

**

***RETURN ON INVESTMENT FORMULATION:***

# **Return on Investment (ROI)**

From an investors point of view, profit matters the most. Hence, formulating profit per prediction by considering the testing accuracy, the cost incurred by a false prediction and expected profit.

Z\_hat = [ P - ( 1 - A ) \* e ]

where,

Z\_hat = Profit per prediction

P = Expected profit

A = Accuracy of the model

e = Cost incurred for every false prediction

(1 - A) gives us the loss generated by our model.

Multiplying it with the cost incurred for a false prediction will give us the cost incurred because of the loss of the model.

Subtracting it from the profit that can be generated from the product will give us the general profit per prediction on using the model.

For our model, we have found the accuracy to be .94 which can be put in place of A.

#### **Estimate**

Let,

P = 100 units

A = .94 (from the model)

e = 150 units (cost incurred by faulty welding)

Therefore,

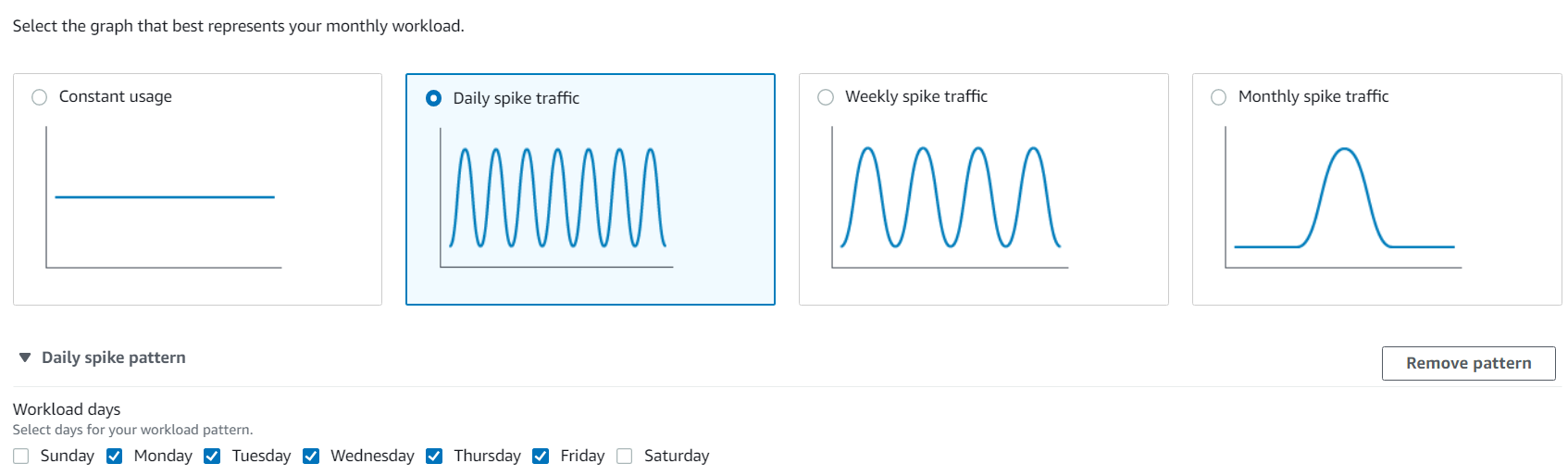
Z\_hat = 91: This means that we can save about 91 units by using the model

# **Total Cost of Ownership**

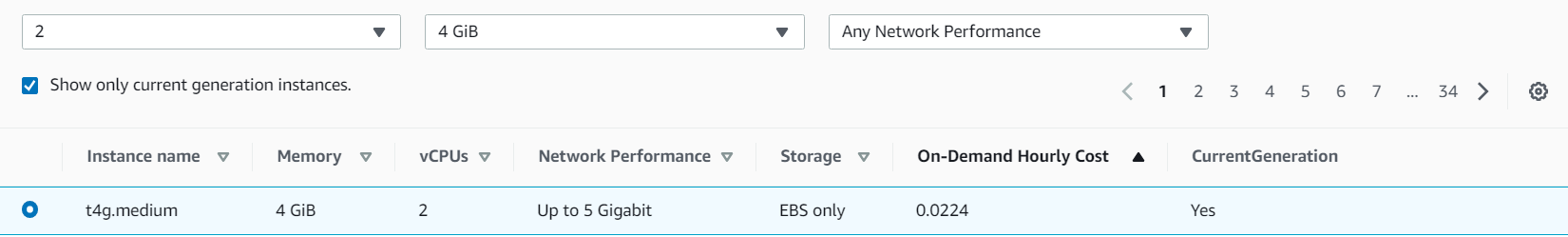
Our model can easily be deployed on cloud and can be accessed anywhere. The best cloud service provider in the industry is Amazon Web Services(AWS) and we will consider all our calculations with respect to the same.

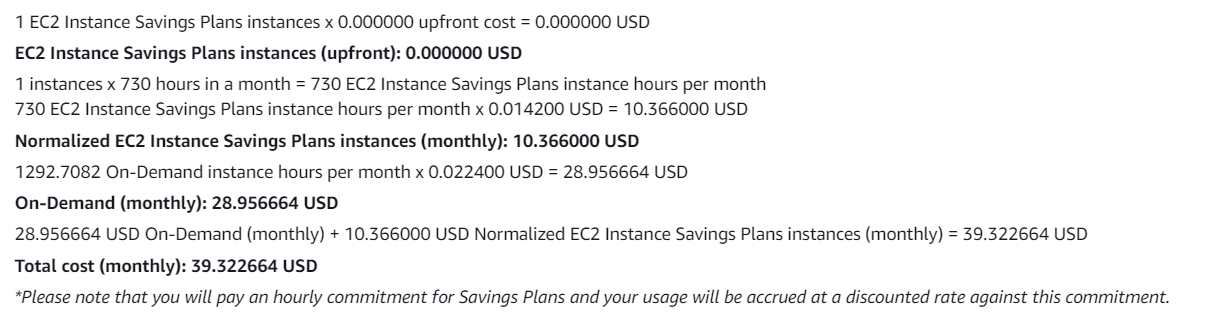
We can estimate the cost of operation of our model by using the tool provided by AWS over here: <https://calculator.aws/#>

1. We have selected the area to be Asia Pacific(Mumbai).
2. We have limited the workload to the working days of the week (Monday - Friday). The initial assumption is that there will be a maximum of 8 instances will be running at the peak of workload

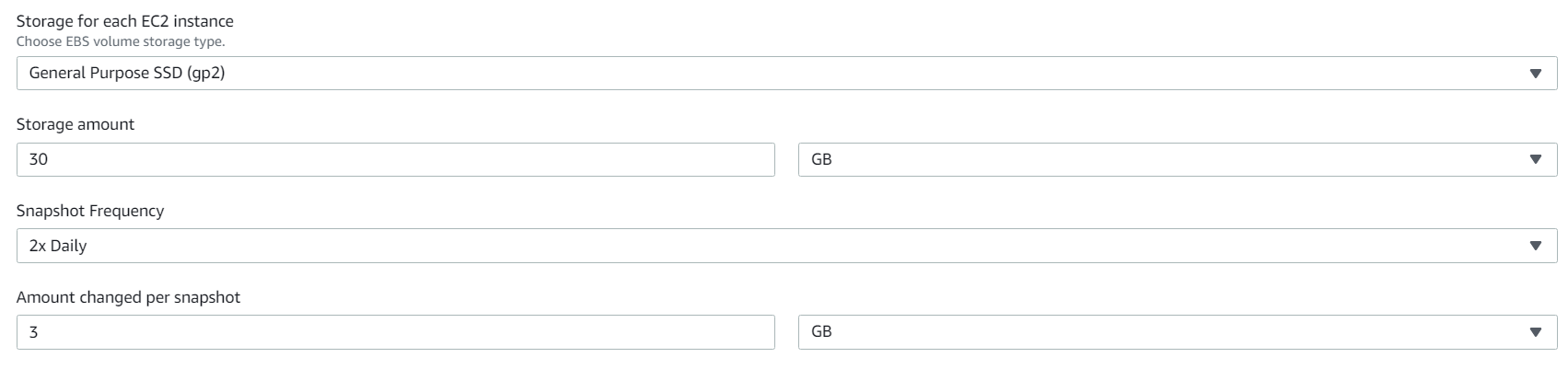


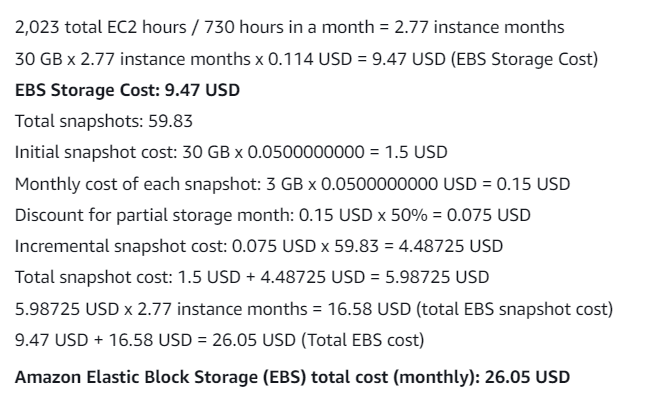
1. We consider 2 vCPUs and 4GB RAM to be enough to carry out computations for predictions using our models. The below image shows the model we are considering.



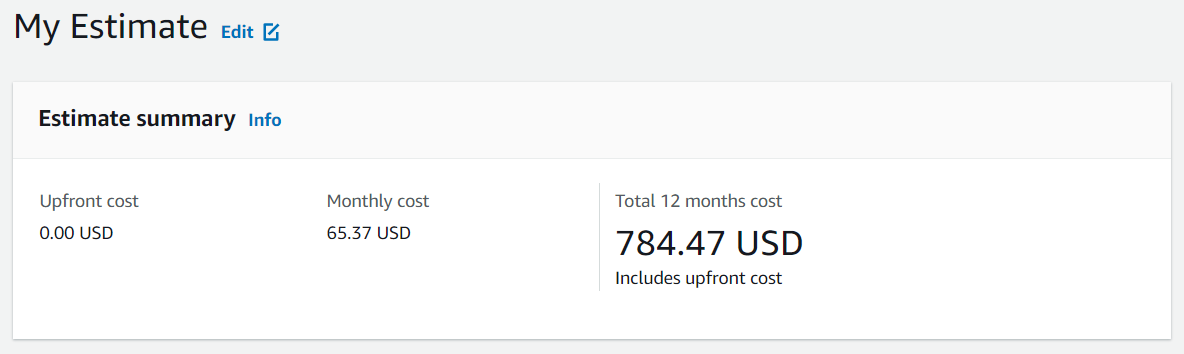
Below is the complete calculation for the monthly cost of using an EC2 service

4.Apart from this, we can also add persistent storage to keep track of the model activity and the entire production. AWS provides Elastic Block Storage(EBS) to keep a track of it. These are the parameters we have chosen.



The monthly calculation for EBS is given below

The Total Cost of Ownership according to our estimation would be :



# **Conclusion:**

1.We have undersampled the majority class and oversampled the minority class using the SMOTE Algorithm for balanacing the input data.

2.We implemented the XGBClassifier Model for prediction with a weighted accuracy of 94%.

3.The approximate annual Total Cost of Ownership is $784.47 based on considering the suitable technical requirements mentioned above.

4.The Return on Investment is formulated above and approximated on the basis of assumptions, we observe that 91 units can be saved per predication(Profit Per Prediction).