DAASA HACKATHON (11-04-23)

TEAM ML MAVERICKS

PROBLEM - I (BAD CYCLE PREDICTION)

TEAM MEMBERS:

1. BHUVVAAN CHANDRA (LEAD)
2. TRINAY GANGISETTY
3. SIDDHARTH KALYANASUNDARAM
4. ANUDEEP NAYAK

PROBLEM STATEMENT:

Objective: Develop a predictive maintenance system for aluminum manufacturing.

Issue: Poor maintenance timing leads to bad furnace cycles, causing costly downtime and low-quality metal.

Solution: Create a model to predict bad cycles and schedule maintenance.

Benefits: Reducing downtime, minimizing maintenance costs, and improving metal quality.

Evaluation: Measure success using AUC-ROC score and Accuracy of the Model.

DATA EXPLORATION (STEP - 1):

1. The first step and the most important step is understanding the columns.
2. We have explored a few insights from the data and they are as follows:
    1. We have identified the target column, quantitative, qualitative variables
    2. We have identified all the columns which had non-numerical text data
    3. We looked into the data set for any missing values
    4. We looked into the statistical analysis for all the numerical columns
    5. We further looked if the data in the columns is normally distributed or if the data is skewed
    6. We have identified that Period and Cycle goes together and cycle starts newly for each period again
    7. We have identified that duration between each cycle is 5 mins
    8. We have identified that there is no overlapping between any two periods and a new period starts only after the previous period is over and not simultaneously.

DATA CLEANING (STEP - 2):

1. **The next step is data cleaning**

   a. Non - numerical text data - > NaN

   b. NaN -> imputed the mean / median values with help of (Kolmogorov - Smirnov) test to check for the "Normality of the data".

   c. If P <= 0.05 -> Median else Mean

   d. We have converted the data types of the numerical columns from object -> float

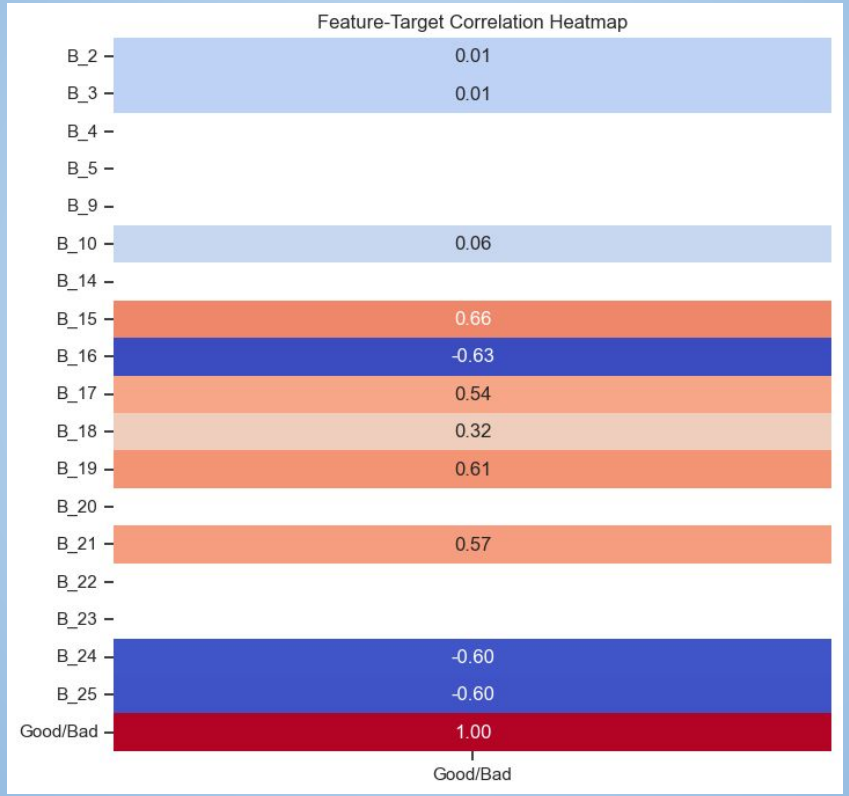
   e. Grouped the values by period -> added +1 to the previous value of missing cycle value and replaced it.

FEATURE ENGINEERING:

We decided to use only a few sensors that are highly correlated and have high variation, and drop the rest of the features. So we explored the same using several methods

1. **RFE - Recursive Feature Elimination**
2. **VIF - Variation Influence Factor**
3. **ANOVA (Analysis of Variance) F - Test**

Correlation Matrix



MODELLING:

ALGORITHM	ACCURACY
RANDOM FOREST CLASSIFIER	91.12%
XGBOOST	95.60%

Why XGBOOST?

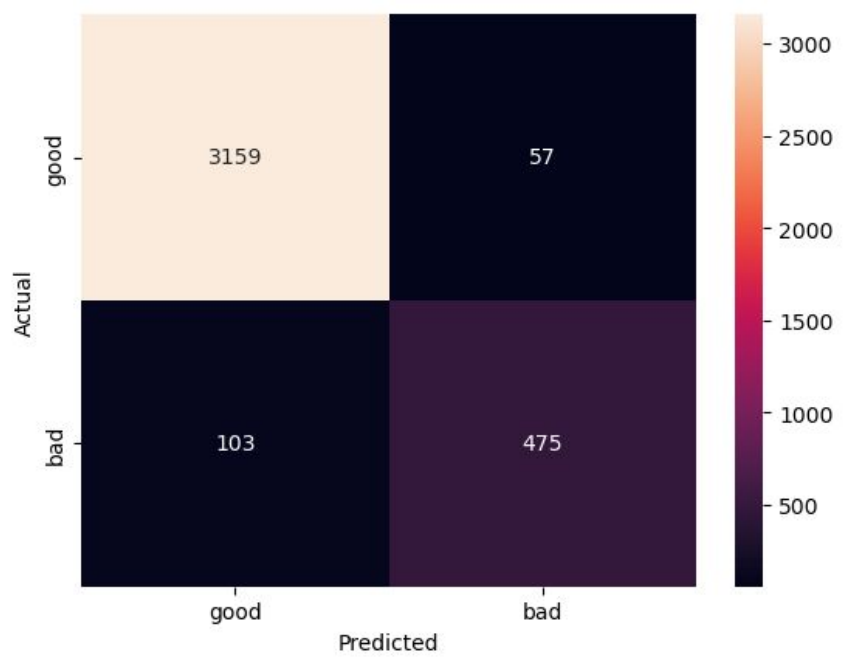
1. **XGBoost is a great algorithm for classification and regression problems.**
2. **It has inbuilt feature selection capability**
3. **Really good for imbalanced datasets such as ours.**
4. **Lots of hyperparameter tuning possibilities.**

Hypertuning:

1. Objective - Logistic
2. Parameter Grid Search
   a. Learning rate - step size learning - 0.02
   b. N_estimators - no of base learners - [150, 300, 450]
   c. Max_depth - height of DT- [3,4,5]

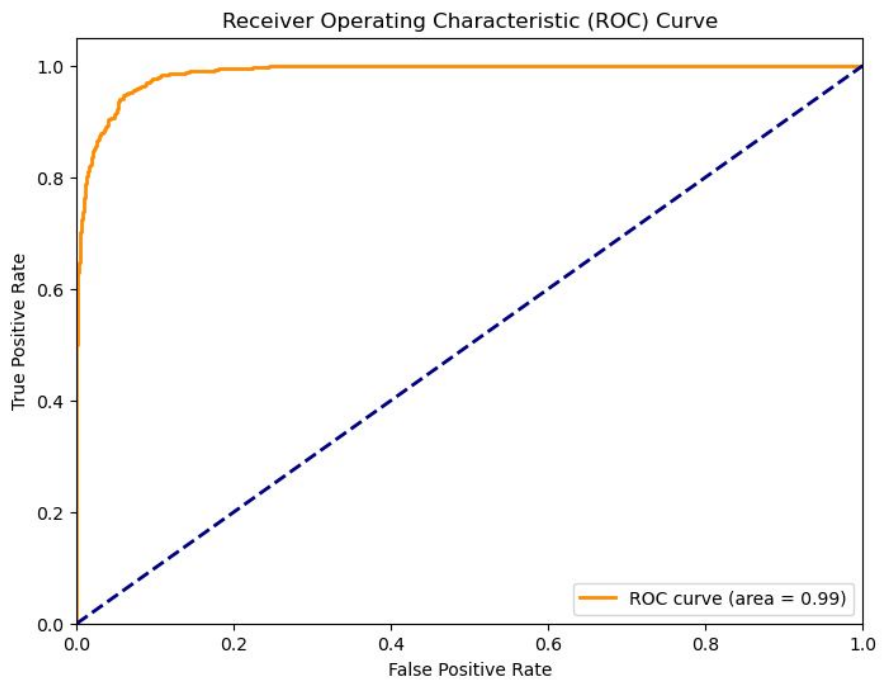
Total 9 possible combinations, 300 and 5 being the best

Confusion Matrix



- The train data set is further split into test and train data sets to estimate the accuracy of the model.
- 75% of the data set is used for training and 25% of the data set is used for testing.
- This leads to a test data set of ~12000.

ROC Curve



- The train data set is further split into test and train data sets to estimate the accuracy of the model.
- 75% of the data set is used for training and 25% of the data set is used for testing.
- This leads to a test data set of ~12000.

Feature Extraction