***Predictive Maintenance of Machinery*** *By Ruchika Sancheti*

Problem Description:

The business problem is about predicting problems caused by component failures such that “What is the probability that a machine will fail in near future due to failure of certain component?” can be answered. The problem is formatted as a multi-class classification problem.

Machine learning algorithms are used to create predictive model that learns from historical data collected from machines.

Objective:

Aim is to create an analytical and modelling framework to predict the future events based on past events for each machine.

The possible major ActionPoints are “ComponentRepair”, “ComponentReplacement” and “NoIssue” in next one month for each machine.

Modelling;

* It is a multi-class imbalance problem.
* Metric for evaluation is F1 score of the lowest represented class “Class Replacement”.
* We need to make sure that model is not biased towards Majority class.
* Several methods/models that handle class imbalance data under scikit-learn module in python

**From my learning, I need two kind of model**

* *Heavily penalize models for wrong predictions –* ***Strong Regularization***
* *Ways to emphasize under represented class –* ***Clustering***

***Model 1 Approach:***

1. **Strong Regularized Model:**

* Linear SVM, SVC, StochasticGradient Classifier

1. **Boosting Techniques:**

* I have implemented AdaBoost and GradientBoosting Models.
* Adaboost gave a decent local F1 score of 33%, but gave around 9% on grader.
* GradientBoosting with hyper parameter tuning didn’t yield any good results either.

1. **Other models trained**

* Built Logistic Regression, Linear SVM, Kernel SVC,

***Model 2 Approach*** : **Emphasize Under represented class**

* I have separated the abundant class (including NoIssue and Component Repair class) as one class (encoder value 0)
* Divided the abundant data into 3 clusters
* Attached minority class (Encoder value 1) rows to all the three clusters
* This way we are tuning models to learn patterns about the minority class.
* Ran Linear SVM, SG classifier and SVC (rbf) on each of the clusters and stored each of the model predictions on test data.
* Merged MachineID and three rows of predictions as one dataframe and derived the result class as mode of three predictor values

|  |  |  |
| --- | --- | --- |
| **Model Name** | **F1 score (Validation data)** | **F1 score (Test data)** |
| **Logistic regression** | **11%** | **12.09%** |
| **Linear SVM** | **18%** | **5%** |
| **SVM (RBF)** | **15%** | **11%** |
| **SG classifier** | **24%** | **20%** |
| **Random Forest (var imp)** | **42%** | **9%** |
| **Adaboost** | **33%** | **9%** |
| **Clustering and Stacking** | **NA** | **15%** |

**Observation**: Clustering from approach 2 and SG classifier from approach 1 are the selected models