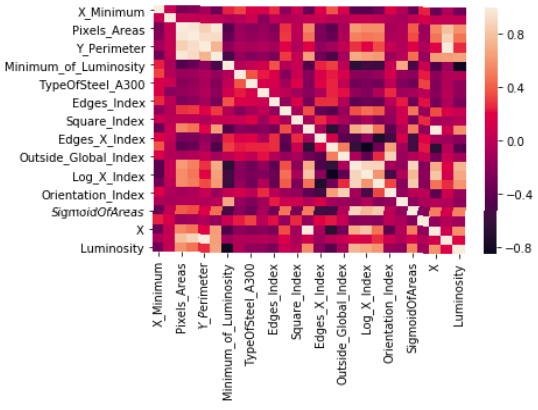
Steel Plate Fault Classification
Abstract

Introduction: A fault may be defined as an unacceptable difference of at least one characteristic property or attribute of a system from acceptable usual typical performance. Therefore, fault diagnosis is the description of the kind, size, location and time of discovery of a fault.

Data Preprocessing: The data set contains 29 features, which determine whether steel plate is faulty. I eliminated those features which where max. and min. Of dimensions and replaced them with the value of range. I removed the Type Of Steel_A400 column as it is redundant due to TypeOfSteel_A300 column. Further on the basis of pearson correlation coefficients, we removed the following features X Perimeter , y Perimeter, Minimum of Luminosity, Orientation Index, Outside Global Index, Luminosity Index and Y. These features were deleted on basis of the below correlation heatmap.



Further, Linear Discriminant Analysis, Principal Component analysis and scaling or normalization were used, but they did not produce desired results so I choose not apply them.

Classification Models: The problem at hand is a binary classification problem. I used the following models with k-fold cross validation: Logistic Regression (84.4%), K - nearest neighbours (85.7%), Support Vector Machine with Linear Kernel (82.9%), Support Vector Machine with rbf kernel (83.2%), Gaussian Naive Bayes (79.1%),Decision tree using C4.5 algorithm with boosting (87.1%) and pruning, Random Forest with 10 trees (94.2%), Multi-Layer Perceptron Neural network with leaky Relu activation functions and sigmoid activation function (88.6%). XG - Boost (Extreme Gradient Boost) with 50 Layers (98.5%) . I came to the conclusion that XGBoost had the best accuracy. Extreme Gradient Boosting (xgboost) is similar to gradient boosting framework but more efficient. It has both linear model solver and tree learning algorithms. Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added in a sequential way until no further improvemen -ts can be made.

The code for the above models has been implemented in Training_Models.py While predicting the output of the test data set I use the same data pre-processing and then used the entire training dataset to fit the model to prevent overfitting.This has been done in Testing.py and the output of test has been saved in Output.csv

Pranjali Sarap
IIT Kharagpur