**Machine Learning Project Report on** **“Machine Failure Prediction”**

**Supervised by: Dr Veena Bansal, IIT Kanpur Rajan Shahurao Bangar (21114015), IME**

**1.Introduction**

In manufacturing industry, heavy machineries are used. These Machines can undergo failures. The operation of a machine **breakdown/Failures** occurs when there is a sudden and unexpected performance decline of a machine. Although failure frequency is very less, but it causes a huge loss to the industry. We will use different methods such as Logistic Regression, K-Nearest Neighbors, Decision Tree classifier, Support Vector Classifier. We will also try to use Neural Network and decide the optimal method, based on assessment of efficiency of classifier algorithm using matrices like accuracy, recall, precision etc.

**2.Problem Definition and Algorithm**

**2.1 Task Definition**

Machine failure data available to us has 19 unique features and 8784 instances as INPUT, in which 2 features are categorical, rest are numerical. OUTPUT is Machine Failure status. Handling imbalanced data is done by trial-error method by using different methods and carefully tuning hyper-parameters.

**2.2 Algorithm Definition**

Various algorithms used are as following:

* Robust Scalar- The input features were not scaled properly and hence were not comparable. This algorithm brought all features between scale of -1 to +1.
* One Hot Encoding- For encoding categorical feature ‘Operator’ into 8 different features with value of 1 for OperatorN & of 0 to other features, if OperatorN is present in given instance.
* Random Under Sampling- Because our original data was imbalanced, we used this algorithm to create a new sub-sample with 2 classes present in 50-50 ratio, by randomly selecting very few instances of abundant class and adding all instances of rare class.
* Over Sampling- Because our original data was imbalanced, we used this algorithm to create a new sub-sample with 2 classes present in, by randomly selecting very few instances of abundant class and adding all instances of rare class.
* Inter-Quartile Range (iqr) method- For handling outliers.
* SMOTE Technique (Over-sampling)-This algorithm created new synthetic points from minority class to reach an equal balance between minority and majority class.
* Cross-Validation- This resampling method uses different subsets of data to train and test the model in different iterations and pick the best of these models.
* t-SNE, PCA, TruncatedSVD - For dimensionality reduction and clustering, we used these algorithms. We used these algorithms to form graphs using which, we can accurately form the clusters of classes in dataset even if the sub-sample is small. These algorithms gave us an indication whether further predictive models will perform well in clustering or not?
* GridSearchCV- We used this algorithm to loop/search through predefined hyper-parameter and fit our model on training set.
* Logistic Regression, K-Nearest Neighbors, Decision Tree classifier, Support Vector Classifier
* Simple Neural Network- We created a simple neural network with three layers using activation functions ‘Relu’ and ‘Softmax’ for building the model.

**3. Experimental Evaluation**

**3.1 Methodology**

Following are the criteria used to evaluate Methods like KNN, SVC etc.:

1. Learning Curves- Used to study Bias & Variance in models.
2. Accuracy score – We used it to calculate number of correct predictions out of total predictions.
3. Recall- This score helped in calculating ability of classifier to find all the positive samples.
4. F1 score- We used it as a combined reflector of precision and recall of a classifier.
5. Confusion Matrix- Tables helped us assess the performance of classification model built.
6. ROC-AUC score- The higher score/ area under curve was desirable as it reflects that our model built is better at predicting 0 class instances as 0 and 1 class instances as 1.
7. Precision Recall Trade off- Models built by principle: precision and recall are inversely related.

**Steps: -** We used data from real situations. Main goal was to build model capable of predicting possible failure of machine. We encoded categorical features into labelled features and then scaled numerical features using robust scalar on scale of (-1 to +1). In dataset, only 81 out of 8784 instances i.e. (0.92%), belong to class 1 reflecting high data imbalance. Hence, we used Random Under-sampling to create a balanced sub-sample of data set. We built models by training on balanced sub-sampled dataset and testing its performance on imbalanced complete dataset. We obtained correlation matrix for sub-sample and removed outliers using boxplot and distribution curve for highly negatively correlated features. Then dimensionality reduction and clustering were done to visualize possible performance of the model. 1)After this pre-processing of input data, we created 4 models using 4 different classifier algorithm, first without cross-validation and then with cross-validation using Under-sampling. Then we created another 4 different models using 4 different classifiers using SMOTE (Over-sampling) method. 2)Lastly, in our study, we created 2 simple Neural Networks first using Under-sampling and then using SMOTE (Over-sampling) methods.

**3.2 Results**

**1. Under-sampling Technique:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Score Metric** | **Method** | **Logistic Regression** | **K Neighbors** | **SVC** | **Decision Tree** |
| **Accuracy without cross-validation(For training)** | | **95%** | **95%** | **96%** | **91%** |
| **Accuracy with cross-validation(For training)** | | **96.32%** | **96.32%** | **97.23%** | **93.55%** |
| **Area under ROC-AUC curve(For training)** | | **0.987** | **0.931** | **0.988** | **0.945** |
| **Recall Score(For testing)** | | **0.92** | **0.97** | **0.97** | **0.89** |
| **Precision Score(For testing)** | | **0.94** | **0.97** | **0.97** | **1.00** |
| **F1 Score(For testing)** | | **0.93** | **0.97** | **0.97** | **0.94** |
| **Accuracy(For testing)** | | **0.95** | **0.98** | **0.98** | **0.96** |

**2. Over-sampling using SMOTE technique:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Accuracy score** | **Precision score** | **Recall score** | **f1 score** | **Average precision-recall score** |
| **0.9273** | **0.2193** | **0.8** | **0.2996** | **0.39** |

**Conclusion:**

1. We should not implement any classifier algorithms directly on imbalanced datasets without pre-processing, because even though the accuracy was more than 99%, but it was result of identification of large number of ‘No Failure’ instances in 99.08% - 0.92% imbalanced dataset.
2. Implementing **Under-Sampling Technique** and using any of 4 classifier methods mentioned in report, the models generated were always better than model created using **SMOTE Over-Sampling Technique**. The reason may be that, “**there are very few Minority class instances, and hence synthetic oversampling of Minority class instances may not be capable of accurately representing Minority class, to help Classifiers build models & accurately identify Minority class instances!”**
3. Implementing **SMOTE (Synthetic Minority Over-Sampling Technique)** on our imbalanced dataset definitely helped us with the imbalance of our labels by identifying more ‘no failures’ than ‘failures’ states. But **this model built was not efficient in identifying ‘Failure’ status instances as much as Under-Sampling method**.
4. Sometimes, neural networks on oversampled dataset predicts less failure states than model using the under-sample dataset. **Reason** may be that, **overfitting could be occurring due to oversampling, which leads to many of failure states wrongly classified!**
5. **Support Vector Classifier** with **Under-Sampling** performed **best** amongst the four classifiers in study. **Reasons** may be that, **large dimension size of data**=19 or **only 2 classes available!**