*Air Pressure System Failures Prediction in Scania Trucks*

Chang Liu   
Gabelli School of Business  
cliu175@fordham.edu

*Abstract*—This is an application-based project with data provided by a Sweden truck company Scania. The dataset consists of data collected from heavy Scania trucks in everyday usage. The system in focus is the Air Pressure System (APS) which generates pressurized air that is utilized in various functions in a truck, such as braking and gear changes. The dataset’s positive class consists of component failures for a specific component of the APS system. The negative class consists of trucks with failures for components not related to the APS. Through this project, I will build several classification models to predict whether it is a truck APS failure or not based on the sensor telemetry data. Also, with a wrong prediction, the cost of unnecessary repairs will increase, so it is required to minimize the false predictions. In this project, I will begin with data preprocessing, solve the problem of data imbalance and data missing, then I will build several models, like logistic regression and random forest. The last step is to compare the performance of different models from the F1-score and cost, based on these we can choose the best model.

Keywords—classification, logistic regression, random forest

# Introduction

The whole dataset includes two parts. The training set contains 60,000 examples in total in which 59,000 belong to the negative class and 1,000 belong to the positive class. The test set contains 16,000 examples. There are 171 attributes per record. Both the training and test set include the class label.

The attribute names of the data have been anonymized for proprietary reasons. The attribute values of these features are numerical counters. The attributes are as follows: class, then anonymized operational data. The operational data have an identifier and a bin id, like "Identifier\_Bin". In total there are 171 attributes, of which 7 are histogram variables. Missing values are denoted by "na".

The total cost of a prediction model is the sum of Cost\_1 multiplied by the number of instances with type 1 failure and Cost\_2 with the number of instances with type 2 failure, resulting in a Total\_Cost. In this case Cost\_1 refers to the cost that an unnecessary check needs to be done by a mechanic at a workshop, while Cost\_2 refer to the cost of missing a faulty truck, which may cause a breakdown. Cost\_1 = 10 and Cost\_2 = 500, as in:

Total\_cost= Cost\_1\*No\_Instances + Cost\_2\*No\_Instances

From the above problem statement, we could observe that we have to reduce False Positives and False Negatives. More importantly we have to reduce False Negatives, since cost incurred due to False Negatives is 50 times higher than the False Positives.

Because we have to reduce False Negatives and False Positives, so clearly we can use precision and recall as performance metric. But here is a simpler metric which takes into account both precision and recall, and therefore, we can aim to maximize this number to make your model better. This metric is known as F1-score, which is simply the harmonic mean of precision and recall.

# ﻿Experiment Methodology

## Data Preprocessing

The project starts with data preprocessing. From the target distribution graph below, we can conclude that the given data is highly imbalanced. In the training set, there are 59,000 samples belong to the negative class and 1,000 samples belong to the positive class So we need to balance the data later.

A screenshot of a cell phone

Description automatically generated

Figure 1: Distribution of Class in Training Data

Also, the attribute values of class column are ‘pos’ and ‘neg’, we replace these with number ‘1’ and ‘0’.

Then we come to the missing value. In the given data, missing values are represented as ‘na’, we will turn ‘na’ to ‘NaN’ format to process the data. From the figure below, we can see there are lots of missing values in the data, so we cannot remove the rows as it will leads to the great amount of information loss.

A screenshot of a social media post

Description automatically generated

Figure 2: Missing Value in Training Data

To avoid the information loss, we will implement below two ideas to deal with the missing values. The first idea is to remove columns which contain more than 75% of ‘NaN’ values, and then, we impute the remaining missing values with mean and median value of that feature. With the help of these two methods, now we have 160 features left.

Now we have two dataframes, one with the mean imputation, and the other with the median imputation. Next we will use SMOTE to balance these two dataframes separately. SMOTE is an oversampling method. It works by creating synthetic samples from the minor class instead of creating copies. The algorithm selects two or more similar instances (using a distance measure) and perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances.

## Random Model

Before we come to the part of model building. At first, we will build a random model as a benchmark for all our models.

A screenshot of a cell phone

Description automatically generated

Figure3: Confusion Matrix of Random Model

Figure 3 above is the confusion matrix and F-1 score of the random model, we can see the F-1 score is 0.045 and the cost is 171,980. Then, we will come to our model building. If the following models we build next has lower matrix than random model, we can ignore that model.

## Data Mining Algorithm

I have built two models with two different dataframes. They are random forest model with mean value, logistic regression model with mean value, random forest model with median value, and logistic regression model with median value. In each model, I split the training dataset into train and test part, I set 70% as training partition and 30% as test patition. And I used the original test dataset to evaluate the performance of each model. When evaluating, I introduced F1-score as a measure to check the performance of the models.

Random forest is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forest corrects for decision trees' habit of overfitting to their training set.

Logistic regression is a statistical method for analysing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables.

The precision-recall curve shows the tradeoff between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall). In this case, we need to pay more attention to False Negatives, since cost incurred due to False Negatives is 50 times higher than the False Positives. Which means, we need to focus more on recall.

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F-1 score is the harmonic mean of the precision and recall, where an F-1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

# Results

As mentioned above, we have built four models, here are the precision and recall curve of these four models.

A close up of text on a white background

Description automatically generated

Figure 4: Curve of Random Forest Model with Mean Value

A close up of a map

Description automatically generated

Figure 5: Curve of Logistic Regression Model with Mean Value

A close up of a map

Description automatically generated

Figure 6: Curve of Random Forest Model with Median Value

A close up of text on a white background

Description automatically generated

Figure 7: Curve of Logistic Regression Model with Median Value

From these 4 figures above, we can see that random forest model with mean value has the highest decision threshold, so we may conclude this model performed best so far. For more details, let’s check the F-1 score and cost.

A screenshot of a cell phone

Description automatically generated

Figure 8: Comparison Table of Models

Figure 8 shows the F1-score and total cost of each model, from this figure, we can see random forest model with mean value has the highest F-1 score, but the total cost of which is high, too. And the random forest model with median value has the lowest cost. So in order to make the decision, we may need to do some further work like building other models or explore the random forest model with mean value for a better performance. For now, we will choose random forest model with mean value as our final model.

# Related work

In this case, I used two methods to impute the missing value, mean and median. Given more time, I’d like to try the imputation with some constant value, like -1. Also, I have built 4 models, but they did not bring much differentiation, so I would like to build more models like XGBoost or SVM. Or try dimensionality reduction techniques like PCA to choose the final model.

# conclusion

Based on the results, we can see random forest model with mean value performed best. However, due to the limitation of number of models and data mining algorithms, there will still be improvements to do.