Machine Learning, Linear and Bayesian Models for Logistic Regression in Failure Detection Problems

B. Pavlyshenko
SoftServe, Inc., Ivan Franko National University of Lviv, Lviv, Ukraine
e-mail: b.pavlyshenko@gmail.com

Abstract— In this work, we study the use of logistic regression in manufacturing failures detection. As a data set for the analysis, we used the data from Kaggle competition “Bosch Production Line Performance”. We considered the use of machine learning, linear and Bayesian models. For machine learning approach, we analyzed XGBoost tree based classifier to obtain high scored classification. Using the generalized linear model for logistic regression makes it possible to analyze the influence of the factors under study. The Bayesian approach for logistic regression gives the statistical distribution for the parameters of the model. It can be useful in the probabilistic analysis, e.g. risk assessment.

Keywords — logistic regression; XGBoost; Bayesian inference; failure detection.

I. INTRODUCTION

It is important to monitor closely the manufacturing parts, as they progress through the manufacturing processes. Bosch records data at every step along its assembly lines. They have the ability to apply advanced analytics to improve these manufacturing processes. However, the intricacies of the data and complexities of the production line pose some problems for current methods. In the Kaggle competition “Bosch Production Line Performance”[1], Bosch is challenging Kagglers to predict internal failures using thousands of measurements and tests made for each component along the assembly line. This would enable Bosch to bring quality products at lower costs to the end user. The peculiarity of this competition is that the classification classes are highly imbalanced. The task of internal failures prediction can be considered as a kind of logistic regression problem. Classifications are evaluated using Matthews correlation coefficient (MCC) between the predicted and the observed response. The MCC is given by

$$ MCC = \frac{(TP \cdot TN) - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} $$

where $TP$ is the number of true positives, $TN$ is the number of true negatives, $FP$ is the number of false positives, and $FN$ is the number of false negatives. The data for this competition represents the measurements of parts, as they move through Bosch's production lines. Each part has a unique Id. The goal is to predict which parts will fail quality control (represented by a Response=1). The data set contains an extremely large number of anonymized features. The goal of this study is to consider different approaches for time manufacturing parts of internal failure modeling. Machine learning algorithms make it possible to find patterns in the data. This approach can give high precision in the logistic regression. We can define the importance of each feature but it cannot give us any information about the influence of each factor. The combination of parametric and non-parametric models can give us more complete analysis with complimentary approaches. As parametric models, we can consider the generalized linear model for logistic regression. Given the features in the analyzed data sets are anonymized, we can consider the internal failures prediction as a stochastic process. If we have sales distribution, we can calculate the value at risk (VaR), which is one of risk assessment features. To find distributions of model parameters Bayesian inference approach can be used.

II. MACHINE LEARNING MODEL

For machine learning approach, we used a gradient boosting classification method implemented in the XGBoost classifier. To perform the analysis, we used R package “xgboost” (short term for eXtreme Gradient Boosting) which is an efficient and scalable implementation of gradient boosting framework [2,3,4]. The package includes efficient linear model solver and a tree-learning algorithm. Our first step was to prepare our data. We merged data frames with numeric, categorical, and date features. We also constructed a new feature denoting the time at which the part was on the manufacturing line. We calculated this feature as the difference between minimum and maximum time in the date features sets. Taking into account the large amount of the data set, we used the undersampling approach
for logistic regression with highly imbalanced classes. We retain the samples with positive response 1 without changes but the number of samples with negative response 0 was reduced essentially using random sampling. For categorical features, one-hot encoding was used. To the data set, obtained in such a way, we applied the XGBoost classifier. The total number of features was more than 4000. We defined the most important features by applying XGBoost classifier to the small subset of the whole training set. For our next investigation, we took 500 top most important features. For the validation data set, we took 25% of training data set samples. The results of the classification were the probability for positive responses. Fig. 1 shows the most important features and their gain values. Fig.2 shows the ROC curve for the classification results. Calculated AUC is 0.753. To get the binary values of response, we need to apply a threshold for received probabilities. If the probability is larger than the threshold, then the response takes on the value 1, otherwise the value is 0. Fig. 3 shows Matthews correlation coefficient for the logistic regression for different values of the threshold.

Fig.1. The most important features and their gain values.

Fig.2. ROC curve for classification results

Fig.3. Matthews correlation coefficient for logistic regression for different values of probability threshold.

### III. GENERALIZED LINEAR MODEL

Logistic regression can be considered as a special case of generalized linear model. Such type of logistic regression enables us to investigate the influence of numeric factors on the binary response. For this model, we can receive the coefficients for each feature under study. The data set under consideration has a large number of not available measure values (NA) for each sample. It is not a big problem for modern machine learning classifiers but when we try to apply the linear logistic regression, we need to preprocess the data to avoid NA values in the data set. The possible reason of such a large number of NA values is that in the data set, there are the manufacturing parts of different types. For one type of the parts, one set of measures is applied for quality controlling, for the other type of the parts, the other set of measures is applied. Taking into account that all measure features in the data set are represented as columns, the measures, which are not applied to some part, will have NA value. So, at first in the linear analysis, we need to group the parts by their types. Grouping can be performed by clusterization. Since the features in the data set are anonymized, we used only numeric features to study the possibility of applying the generalized linear logistic regression for this data set. We used the k-mean algorithm for the clusterization. We set up the value 0 for not available values of features in the data sets and value 1 for the features with numeric values of measure. To speed up the calculation, we chose 850 random features. Fig.3 shows the dependency of total within-clusters sum of squares from the number of clusters. The results show that the optimal number of groups of the parts is approximately 20-30.
We took the value 25 for the number of clusters. For the linear regression, we took the parts from one arbitrary cluster only. We removed the columns where there were more than 10% of NA values. We also removed the rows with more than 10% of NA values. The rest NA values were replaced by median value, calculated for appropriate numeric features. For the linear logistic regression, we used R package “glmnet” [5,6,7,8]. We used the logistic regression with LASSO regularization. The glmnet algorithms use cyclical coordinate descent, which optimizes successively the objective function over each parameter with the others fixed, and cycles repeatedly until the convergence [8]. Let us denote the probability that if the response is 1 by $p_1$, then the probability that the response is 0 will be $p_0=1-p_1$. For the logistic regression, we can write

$$\log\left(\frac{p_1}{p_0}\right) = \beta_0 + \beta^T x$$

$$p_1 = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)}$$

The coefficients $\beta, \beta^T$ can be found via minimizing the appropriate objective function. To find the optimal value for regularizing the Lambda parameter, we used the cross validation approach. Fig.5 shows the dependence of logarithmic value of Lambda from AUC value. The axis above indicates the number of nonzero coefficients for current Lambda, which is the effective degree of freedom for the LASSO model. Having this dependence, we can find the optimal value of regularization parameter, under which we can receive the best fitting.

Using Lambda=0.03, we calculated nonzero coefficients of the generalized linear model for logistic regression which is shown on Fig.6.

Fig. 7 shows the histograms, correlation coefficients, pairs scatterplots for some chosen features. This figure shows the character of dependencies between these features.
IV. BAYESIAN MODEL

For Bayesian model, we took the features which were found in the generalized linear model using LASSO regularization. The analysis was conducted using JAGS sampler software with “rjags” R package [9,10]. For modeling, we used the logistic regression. The idea is that a linear combination of metric predictors is mapped to the probability value via the logistic function, and the predicted 0 and 1 are Bernoulli distributed [9]. We can describe logistic regression as

\[ p = \text{Logistic}(b_0 + b_1 x_1 + b_2 x_2 + ... b_n x_n) \]

\[ y \sim \text{Bernoulli}(p) \]

where \( \text{Logistic}(x) = \frac{1}{1 + \exp(-x)} \)

The simple probabilistic model for logistic regression using BUGS syntax has the following code:

```r
model{
  for (i in 1:n) {
    y[i] ~ dbern(p[i])
    logit(p[i]) <- b0+inprod(b[ ],x[i,])
  }
  b0 ~ dnorm(0,0.0001)
  for (j in 1:nfeat) {
    b[j] ~ dnorm(0,0.0001)
  }
}
```

Trace plots of samples vs the simulation index can be very useful in convergence assessment. Fig. 8 shows the trace plot for \( b_0 \) parameter of logistic model.

The trace plot demonstrates the stationary process, which means good convergence. Fig. 9 shows the probability density distributions for Intercept \( (b_0) \) parameter.

Fig. 10 shows the examples for box plots for some logistic regression coefficients. The points on the figure denote the values of coefficients calculated using generalized linear model without regularization \( (\Lambda=0) \) which was applied to the same features as in the case of Bayesian model.
V. CONCLUSION

In our case study, we showed different approaches for the logistic regression in the problem of manufacturing failures detection. Machine learning approach can give the best-scored failure detection. The generalized linear model for the logistic regression makes it possible to investigate the influence factors on the failure detection in the groups of manufacturing parts. Using Bayesian model, it is possible to receive statistical distribution of model parameters, which can be used in the risk assessment analysis.

[1] Kaggle competition “Bosch Production Line Performance”. URL: https://www.kaggle.com/c/bosch-production-line-performance
[2] Chen, Tianqi, and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” arXiv preprint arXiv:1603.02754 (2016).
[3] J. Friedman. “Greedy function approximation: a gradient boosting machine.”, *Annals of Statistics*, 29(5):1189–1232, 2001.
[4] J. Friedman. “Stochastic gradient boosting.”, *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.
[5] Jerome Friedman, Trevor Hastie and Rob Tibshirani. (2008). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, Vol. 33(1), 1-22 Feb 2010.
[6] Noah Simon, Jerome Friedman, Trevor Hastie and Rob Tibshirani. (2011). Regularization Paths for Cox’s Proportional Hazards Model via Coordinate Descent. *Journal of Statistical Software*, Vol. 39(5) 1-13.
[7] Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, Ryan J. Tibshirani. (2010). Strong Rules for Discarding Predictors in Lasso-type Problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2), 245-266.
[8] Stanford Statistics Technical ReportGlmnet Vignette . URL: http://www.stanford.edu/~hastie/glmnet/glmnet_alpha.html
[9] Kruschke, John. *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press, 2014.

Fig. 10. Box plots for logistic regression coefficients.