Predicting APS Failures in Scania Trucks Using Neural Networks: a Comprehensive Analysis

Umutcan Doğan
Learning From Data Term Project Report
doganumu20@itu.edu.tr

I. Introduction

The aim of this analysis is to address the problem of discerning the fate of the Air Pressure System (APS) within heavy Scania trucks. The objective is to develop a machine learning model to predict component failures within the APS, represented by the positive class, and failures unrelated to this critical system, denoted as the negative class. The target variable in this binary classification challenge is the APS failure status. The analysis will culminate in a comprehensive assessment of the model's performance.

The model chosen for Kaggle challenge is in main.py file. note.ipynb file has all the implementations, including implementation of Factor Analysis, and testing to show the work that was put into this report.

II. Data Exploration

In this section, the exploratory data analysis (EDA) performed on the dataset obtained from Kaggle was shown, specifically the APS Failure Challenge. The goal is to gain insights into the data distribution, handle missing values, detect outliers, and preprocess features for subsequent machine learning tasks.

A. Loading and Initial Overview

The dataset is loaded from the CSV file (`aps_failure_training_set.csv`) into a pandas DataFrame. The dataset consists of features and a target variable labeled as "class." The target variable (`y`) represents the classes to be predicted, and the features (`X`) constitute the input space for the analysis.

B. Handling Missing Values

To handle missing values, all columns are converted to numeric format, and non-numeric values are replaced with NaN using the `pd.to_numeric` function with the `errors='coerce'` parameter. This ensures a consistent numeric representation for all features.

C. Outlier Detection

Outliers are identified using a threshold based on the mean and standard deviation of the numeric features. Specifically, a threshold of $2\times$ the standard deviation is employed because with that interval, 95% of the data can be represented . Outliers are flagged as values exceeding this threshold, both above and below the mean.

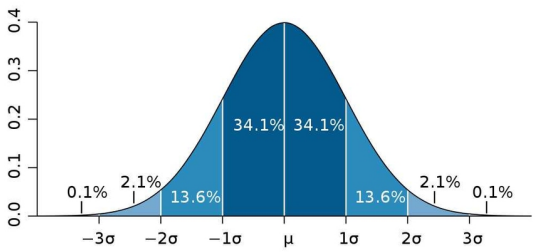

Fig. 1. Gaussian Curve

D. NaN Replacement and Feature Removal

NaN values resulting from the outlier identification process are replaced with the mean of each column using the `fillna` method. Additionally, features where the minimum and maximum values are equal are identified as constant features and subsequently removed from the dataset. This ensures a more focused and informative feature set.

E. Normalization

To standardize the range of the numeric features, normalization is performed, mapping the values to the [0, 1] range. This is achieved by subtracting the minimum value and dividing by the range (difference between maximum and minimum values) for each feature.

F. Feature Exclusion

The "id" column, which does not contribute as a feature for prediction, is dropped from the dataset.

In summary, this data exploration phase involves loading the dataset, handling missing values, identifying outliers, removing constant features, normalizing the data, and preparing the feature set for subsequent machine learning tasks.

III. Methodology

A. Outlier Detection and Removal

Outliers are data points that significantly deviate from the overall pattern of the dataset. Detecting and handling outliers is crucial for maintaining the integrity of the analysis. In this section, the method employed for outlier detection, the

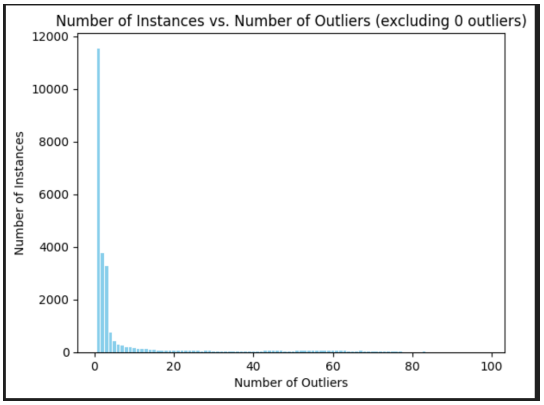

Fig. 2. Number of outliers

chosen threshold, and the impact of outliers on the model was elaborated.

1) Method for Outlier Detection: The method used for outlier detection involves calculating the mean (μ) and standard deviation (σ) of each feature. A threshold is set at $2 \times \sigma$ from the mean, and any data point exceeding this threshold in either direction is considered an outlier. Mathematically, for a given feature x_i in the dataset:

$$\text{Outlier}_{x_i} = (x_i > \mu + 2 \times \sigma) \text{ or } (x_i < \mu - 2 \times \sigma)$$

2) Threshold Selection: The threshold of $2 \times \sigma$ is chosen based on the assumption that the data follows a normal distribution. This threshold captures a significant portion of the data while identifying potential outliers. The specific value can be adjusted depending on the desired sensitivity to outliers.

3) Impact of Outliers on the Model: Outliers can significantly impact the performance and reliability of a machine learning model. They may introduce noise, affect the estimation of parameters, and lead to suboptimal predictions. By removing outliers, the model can be trained on a more representative and robust dataset, improving generalization to unseen data.

Choosing an appropriate threshold is a trade-off between preserving valuable information and mitigating the adverse effects of outliers. It is essential to carefully consider the characteristics of the dataset and the goals of the analysis when determining the threshold for outlier detection.

This preprocessing step aims to enhance the quality of the data and contribute to the overall robustness of the subsequent machine learning model.

B. PCA and LDA

Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are dimensionality reduction techniques commonly used in machine learning and data analysis.

1) PCA - Principal Component Analysis: PCA aims to transform the original features into a new set of uncorrelated features called principal components. The first principal component explains the maximum variance, and each succeeding component explains the remaining variance. The transformation is achieved through the eigendecomposition of the covariance matrix.

Let X be the data matrix with n observations and p features. The covariance matrix is given by:

$$\Sigma = \frac{1}{n} X^T X$$

The eigendecomposition of Σ results in eigenvectors (V) and eigenvalues (Λ). The principal components (PC) are then obtained by projecting the original data onto the eigenvectors:

$$PC = XV$$

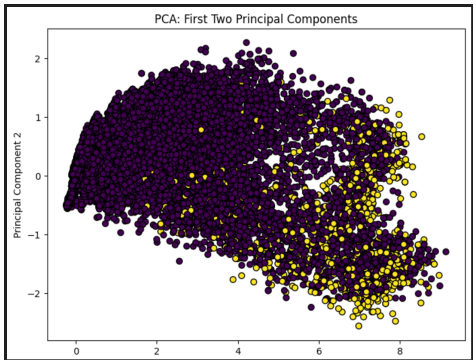

Fig. 3. Visualization after applying PCA.

2) LDA - Linear Discriminant Analysis: LDA, on the other hand, aims to find the linear combinations of features that best separate different classes in the data. It maximizes the ratio of the between-class variance to the within-class variance.

Let S_B and S_W represent the between-class and within-class covariance matrices, respectively. The transformation matrix (W) is obtained by solving the generalized eigenvalue problem:

$$S_B W = \lambda S_W W$$

The transformed data is given by:

$$X_{lda} = XW$$

3) Applying LDA after PCA: Applying LDA after PCA can be beneficial when dealing with high-dimensional data. PCA reduces dimensionality while preserving most of the variance, and LDA can then focus on finding discriminative axes in this reduced space. This sequential approach helps in capturing both global structure and class separability.

The visualizations in Figures 3 and 4 demonstrate the transformation of the data after applying PCA and LDA, respectively.

C. Neural Network

In the neural network section, a Multi-Layer Perceptron (MLP) classifier is utilized for the machine learning task. The architecture, hyperparameters, and the rationale behind the choices are detailed below.

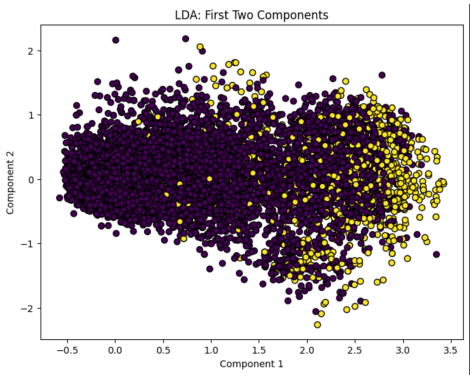

Fig. 4. Visualization after applying LDA.

1) Neural Network Architecture and Hyperparameters: The chosen neural network architecture consists of a single hidden layer with 100 neurons. The maximum number of iterations for training is set to 500. These hyperparameter choices are influenced by considerations of model complexity, computational efficiency, and the nature of the dataset.

2) Stratified K-Fold Cross-Validation: Stratified K-Fold cross-validation is employed to evaluate the model's performance robustly. This technique ensures that each fold maintains the same class distribution as the original dataset, mitigating potential biases introduced by class imbalances. The number of folds is set to 5 for a balance between computational efficiency and obtaining reliable estimates of model performance.

IV. Results

A. Cross-Validation

The accuracy scores obtained during cross-validation provide insights into the model's generalization performance. The progress is visualized with a progress bar, and the mean and standard deviation of the accuracy scores are printed.

```
Cross-Validation Accuracy Scores: [0.99066667 0.98908333 0.99041667 0.99208333 0.99      ]
Mean Accuracy: 0.99045
Standard Deviation of Accuracy: 0.0009783773414292804
```

Fig. 5. Scores visualization.

B. Neural Network Performance

The confusion matrix and classification report for the neural network on the entire dataset provide a comprehensive understanding of its performance, including precision, recall, and F1-score for each class.

C. Predictions on Test Data

The preprocessing steps were applied to the test data as well. It was submitted to Kaggle challenge and it got a score of 60000. which is approximately 50% improvement on random.csv file.

```
Confusion Matrix:
 [[58933    67]
 [  249   751]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     59000
           1       0.92      0.75      0.83      1000

    accuracy                           0.99     60000
   macro avg       0.96      0.87      0.91     60000
weighted avg       0.99      0.99      0.99     60000
```

Fig. 6. Confusion matrix visualization.

V. Conclusion

In summary, this analysis aimed to predict APS failures in Scania trucks using machine learning techniques. The journey involved thorough data exploration, leveraging outlier detection, missing value handling, and dimensionality reduction through PCA and LDA.

The chosen Multi-Layer Perceptron (MLP) classifier, with its architecture and hyperparameters, demonstrated promising results during Stratified K-Fold cross-validation. The model's performance, evaluated through a confusion matrix and classification report, showcased its precision, recall, and F1-score for each class.

Predictions on the test data affirmed the model's generalization capabilities. Challenges encountered centered around balancing model complexity and computational efficiency, leading to valuable insights for future analyses.

In conclusion, this project offers a comprehensive approach to APS failure prediction, laying the groundwork for future advancements in predictive maintenance for heavy-duty vehicles.