Machine Learning Engineer Nanodegree

Capstone Proposal – Truck Failure Predictions

Barbara Coleman
February 22, 2018

Proposal

Domain Background

Scania Trucks, in Stockholm, Sweden, uses a system called *APS Failure and Operational Data for Scania Trucks* to track multiple data points from their trucks and to record mechanical failures, specifically failures related to the Air Pressure System (APS).

The dataset consists of data collected from heavy Scania trucks in everyday usage. The system in focus is the Air Pressure System which generates pressurised air that are 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. The data consists of a subset of all available data, selected by experts.

I found this dataset on the UCI Machine Learning Repository. This dataset was used during the Industrial Challenge 2016 at the 15th International Symposium on Intelligent Data Analysis (IDA). I am interested in this particular dataset, as it is a real-world problem related to the Supply Chain side of business. I have significant work experience with both wholesale and retail Supply Chain operations.

Problem Statement

The ability to predict failures, based on collected data from the trucks, before the failure happens would reduce maintenance costs.

The cost of a False Negative is 50 times higher than the cost of a False Positive. See details below in the Evaluation Metrics section. In this case Cost_1 (False Positive) refers to the cost that an unnecessary check needs to be done by an mechanic at a workshop, while Cost_2 (False Negative) refer to the cost of missing a faulty truck, which may result in a breakdown.

Datasets and Inputs

The training set contains 60000 examples in total in which 59000 belong to the negative class and 1000 positive class. The test set contains 16000 examples. The full dataset can be found here.

The attribute names of the data have been anonymized for proprietary reasons. It consists of both single numerical counters and histograms consisting of bins with different conditions. Typically, the histograms

have open-ended conditions at each end. For example, if we measuring the ambient temperature 'T' then the histogram could be defined with 4 bins where:

bin 1 collect values for temperature T < -20
bin 2 collect values for temperature T >= -20 and T < 0
bin 3 collect values for temperature T >= 0 and T < 20
bin 4 collect values for temperature T > 20

b1	b2	b3	b4
-20	0	20	

The attributes are as follows:

- Class (pos or neg)
- 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'.

Solution Statement

I will be attempting to use a Neural Network algorithm on this supervised, binary classification problem, partly due to the large number of features available and partly due to the fact that I would like to increase my experience with Neural Networks.

My goal will be to get a result that is better than the two “naïve” assignment scores, and to attempt to come close to the top three scores from the 2016 competition which targeted this same problem. (*as noted in the Benchmark Model section below*)

Benchmark Model

There are three benchmarks that will be used for comparison purposes during this project.

1) The naïve “always negative” assignment
    a. Total Cost Score: 188,000 (Type 1 Faults = 0; Type 2 Faults = 376)
2) The naïve “always positive” assignment
    a. Total Cost Score: 156,260 (Type 1 Faults = 15,626; Type 2 Faults = 0)
3) Comparison to the top 3 winners of the 2016 Industrial Challenge[1] on this problem, which were:
    a. Total Cost Score: 9920 (Type 1 Faults = 542; Type 2 Faults = 9)
    b. Total Cost Score: 10900 (Type 1 Faults = 490; Type 2 Faults = 12)
    c. Total Cost Score: 11480 (Type 1 Faults = 398; Type 2 Faults = 15)

[1] *as reported in the Overview document from the UCI Machine Learning Repository*

Evaluation Metrics

The cost of a False Negative is 50 times higher than the cost of a False Positive. Therefore, the Cost Metric is defined as follows:

	True class	
Predicted class	*Pos*	*Neg*
Pos	--	Cost_1
Neg	Cost_2	--

Where: Cost_1 = 10 and Cost_2 = 500

Therefore, 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'.

Total_cost = Cost_1*# False Positives + Cost_2*# False Negatives

Project Design

Tools and Libraries:

Python, Jupyter Notebooks, pandas, scikit-learn, Keras, tensor flow. Other libraries will be added as required.

Pre-Processing:

Data Preparation will include replacing missing ("na") values with appropriate median or mean values.

Some feature engineering may be done over the 7 histogram (binned) variables. These may include Sum of Bins, or Histogram Distance Metrics (ie: Earth Mover's Distance[2])

Feature selection may be done to reduce the total number of input features, if that appears to be beneficial for accuracy and/or processing time. Where we are starting with 170 features, and may add additional engineered features for the histograms, it is worth exploring this requirement. Various feature selection methodologies may be used, including Filter, Wrapper and Embedded; and using either a Random Forest[3] algorithm or a basic Neural Network model for comparisons.

If any features require it, a normalization process will be applied.

Modelling:

Generate a simple Neural Network model using a couple of Dense layers as the starting point. Additional or different types of layers (Pooling, Dropout, etc.) and tweaking of parameters will be tried in an attempt to improve the accuracy based on the defined scoring metric.

---

[2] Rubner, Y., Tomasi, C., Guibas, L.J.: A Metric for Distributions with Applications to Image Databases. In: Proceedings of the Sixth International Conference on Computer Vision. pp. 59–. IEEE Computer Society, Washington, DC, USA (1998)

[3] Random Forest was used by one of the top 3 finishers in the Industrial Challenge 2016, as published by Gondek, C., Hafner, D. and Sampson, O.