Nanodegree, Machine Learning Engineer, Capstone Project Proposal

Fernando Martinelli Ramacciotti[1]

I. Introduction

This proposal aims to motivate the Capstone Project of Udacity's Machine Learning Engineer Nanodegree about *Predictive Maintenance*. I intend to briefly introduce the proposed theme by walking through seven key aspects that any end-to-end machine learning project should cover: (i) Domain Background; (ii) Problem Statement; (iii) Data; (iv) Proposed Model; (v) Benchmark; (vi) Evaluation Metrics; and, finally, (vii) Workflow.

II. Domain Background

Business are always interested in cost reduction opportunities, since it may allows greater profits. Traditionally, companies schedule maintenance of their assets on a regular basis and assess then the need for replacement or fixing. In short, companies rely on corrective maintenance and actions are only taken when the asset is already degraded or failed. Predictive Maintenance, thus, aims to accurately provide remaining useful life of assets so that businesses can prepare in advance when such asset comes to fail. Such prepared action can be replacement, maintenance, money borrowing and any other strategy that will minimize the loss of the to be faulty asset in question.

III. Problem Statement

This project aims to predict the remaining useful life (RUL), in number of cycles, of turbofans engines at any given time.

IV. Data

The dataset used is from NASA's repository: Turbofan Engine Degradation Simulation Data Set and can be found in `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository`. The data is described as follows (taken from the description that comes with the data).

Data sets consists of multiple multivariate time series. Each data set is further divided into training and test subsets. Each time series is from a different engine, i.e., the data can be considered to be from a fleet of engines of the same type. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user. This wear and variation is considered normal, i.e., it is not considered a fault condition. There are three operational settings that have a substantial effect on engine performance. These settings are also included in the data. The data is contaminated with sensor noise.

The engine is operating normally at the start of each time series, and develops a fault at some point during the series. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure. The objective of the competition is to predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate. Also provided a vector of true Remaining Useful Life (RUL) values for the test data.

The data are provided as a zip-compressed text file with 26 columns of numbers, separated by spaces. Each row is a snapshot of data taken during a single operational cycle, each column is a different variable. The columns correspond to:

- unit number;
- time, in cycles;
- operational settings (3 columns);
- sensor measurements (21 columns).

We have four different files:

- FD001: 100 train samples and 100 test samples, one operational condition and one fault mode;
- FD002: 260 train samples and 259 test samples, six operational conditions and one fault mode;
- FD003: 100 train samples and 100 test samples, one operational condition and two fault modes;
- FD004: 248 train samples and 249 test samples, six operational conditions and one fault modes;

V. Proposed Approach

It can be treated as classification or regression problem:

- Classification: the dataset could be labeled in a fashion that allows the model to predict the probability to failure in the next X units of time. Therefore, the X points preceding a known failure belongs to a class *failure* and the rest *normal*. If more than one failure is present, than we have a multi-classification problem at hand;
- Regression of any kind.

Note that when treated as classification, the problem statement is slightly different than stated in Section III, since it does not provide a forecast of when a failure will occur but only if it would be on the next X periods of time.

VI. Benchmark

The benchmark result of this problem will be the results of a simples model, such as logistic regression (for classification) and linear regression (for regression).

[1] `fernandoramacciotti@gmail.com`

VII. Evaluation Metrics

Treated as a classification problem we can use a combination of accuracy, precision, recall, F1 scores and cost adjusted ROC (receiver operating characteristics). For regression we could use mean absolute error, mean squared error or R^2. Also, cross-validation is often a good practice to avoid overfitting and to better tune the hyperparameters of the chosen model(s).

VIII. Workflow

The proposed workflow is structured as follows:

A. Data split

I would hold the test set only for testing and split the training part in training and validation.

B. Exploratory Data Analysis

With the final training set, I would plot the data in order to uncover and understand patterns. Such patterns are important to assess the need of feature engineering (i.e. create new calculated variables). Descriptive statistics also plays an important role in order to be familiar with variables' distributions and eventual missing data. Such step is crucial in order to detail the pre-process the data that will feed the model(s).

C. Model selection and tuning

Multiple algorithms and statistical models will be tested and tuning using the validation set. The comparison with the benchmark model is important to assess if increased complexity reflects increased performance.

D. Test

After the validation phase, the models are assessed using the test data.

References

[1] Saxena, Abhinav, et al. ”Damage propagation modeling for aircraft engine run-to-failure simulation.” Prognostics and Health Management, 2008. PHM 2008. International Conference on. IEEE, 2008.

[2] https://docs.microsoft.com/en-us/azure/machine-learning/desktop-workbench/scenario-predictive-maintenance

[3] https://www.svds.com/predictive-maintenance-iot/