Predictive Maintenance using Machine Learning on Sagemaker

Introduction

Imagine you are the manager at a manufacturing company responsible for monitoring assembly lines. Each assembly line contains multiple kinds of machines that must work continuously and reliably to produce ready-to-ship products as can be seen in the image below. IoT sensors placed on these machines monitor electricity consumption, noise generated, vibration, temperature and various other measurable quantities that are used to monitor the health of each machine. Sudden breakdown of any of these machines across multiple assembly lines will lead to:

1. unscheduled downtime and resulting delays in delivering your product to market
2. cost incurred due to delays, and hiring maintenance workers to repair and possibly replace parts of the machine that caused the breakdown

< *From https://www.raconteur.net/manufacturing/how-to-increase-efficiency-in-production*

You have been tasked with researching a technique called "**predictive maintenance**", especially after your competitors Advantech, Inc. have published a report (http://www.advantech.com/industrial-automation/industry4.0/pms#my_cen). Additionally, you are intrigued to see if Machine Learning can help with this problem. Your team's collective research notes regarding a potential proof-of-concept that you will be building is included here:

Reactive, Predictive or Preventive Maintenance

Maintenance schedules in typical manufacturing and energy companies that involve large number of machines performing tasks are typically a result of "reactive" or "preventive" maintenance. A reactive maintenance task is scheduled if a machine breaks down (or fails), or is operating in a known degrade state of operation. Preventive maintenance is triggered by usage or time or a fixed schedule. As an example, for car owners, reactive maintenance occurs after there is an failure of a component (stalled engine, punctured tire, etc.), whereas preventive maintenance occurs on a fixed schedule (for example, tire rotation or oil change every 10000 miles) even though there may not be a need for doing so.

"Predictive" maintenance implies that maintenance is scheduled when a system predicts the possible occurrence of a failure even in the future. This solves problems that are common in reactive and preventive maintenance - 1. reactive maintenance adds unnecessary time and cost to project schedules, since maintenance workers are deployed only *after* discovery of a failure event; and 2. preventive maintenance adds unnecessarily frequent maintenance tasks, therefore increasing wait times and costs for the end user. Currently, predictive maintenance techniques involve simple monitoring-and-thresholding, or statistical techniques to identify anomalies from sensor data. However, these techniques are limited to use by Subject Matter Experts (SMEs) and depend on human-generated thresholds. With Machine Learning (ML), it is possible to train models to detect abnormal patterns from sensor data. The trained ML model does not require rules or pre-programmed thresholds, and vast amounts of data can be analyzed repeatably with no need of human involvement.

NASA Turbofan Engine Fault Dataset

BACKGROUND

NASA's Prognostic Center of Excellence established a repository with datasets to be used for benchmarking prognostics and predictive maintenance related algorithms. Among these datasets involves data from a turbofan engine simulation model C-MAPPS (or Commercial Modular Aero Propulsion System Simulation). The references section contains details about the over 100 publications using this dataset. C-MAPPS is a tool used to generate health, control and engine parameters from a simulated turbofan engine. A custom code wrapper was used to inject synthetic faults and continuous degradation trends into a time series of sensor data. Some high level characteristics of this dataset are as follows:

1. The data obtained is from a high fidelity simulation of a turbofan engine, but closely models the sensor values of an actual engine.
2. Synthetic noise was added to the dataset to replicate real-world scenarios.
3. The effects of faults are masked due to operational conditions, which is a common trait of most real world systems.

Download the dataset here - https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#turbofan.

*** s3 sync link instead of downloading from NASA

MORE DETAILS

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 ñ 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:
1) unit number
2) time, in cycles
3) operational setting 1

4) operational setting 2
5) operational setting 3
6) sensor measurement 1
7) sensor measurement 2
...
26) sensor measurement 26

(From the Readme included in the dataset zip)

OTHER USEFUL POINTERS SPECIFIC TO THIS ML PROBLEM

Predictive maintenance using the turbofan engine dataset has been thought of as one of the following ML problems:

1. Classification problem -
    a. predict if the engine will fail in a particular time window (yes / no)
    b. predict which windows the engine will fail in (e.g., window 0, window 3, and window 4)
2. Regression problem -
    a. predict Time to Failure (TTF)
    b. predict Remaining Useful Life (RUL)

The data provided is not in a format suitable for all of the above ways of thinking about the predictive maintenance ML problem. How will you modify the data to suit your model?

Are all features important? Can some features be combined or aggregated?

Are your classes (if you have any) evenly distributed? If not, how do you help balance this class distribution?

How do you evaluate the accuracy of your model? What cells in the confusion matrix (if you have one) have higher impact for a fictitious turbofan engine manufacturer that uses your model?

References

https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/publications/#turbofan

https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150007677.pdf

A. Saxena, K. Goebel, D. Simon, and N. Eklund, “Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation”, in the Proceedings of the Ist International Conference on Prognostics and Health Management (PHM08), Denver CO, Oct 2008.