Springboard Data Science Capstone Project - Predicting Machine Failures

Ramaa Vissa

Feb 10, 2020

Contents

1. Introduction

Most manufacturing factories have a quality control unit that oversees the predictive maintenance. On the floor, machine failures can occur due to wear and tear of the machine over prolonged use. The problem is to capture the failed unit before it can cause more failures downstream.

The goal of this project is to predict the failed unit.

The data presented here is from a quality station of a manufacturing plant for one device.
There are 219 parameters measured for each device.
Machine_State column provides information on whether the machine passed or failed the quality checks.

We would like to implement a machine learning algorithm to detect a bad device using the available 219 parameters. In addition to the above information, one other important information is the cost function - The cost of a bad device passed in the station which fails in the field is $5000 and the cost of testing a good device classified as bad is $500.

2. Data Acquisition and Cleaning

We acquire data sets from the quality control department. The department has provided machine logs that capture several features.

1. Telemetry dataset

The first data source is the telemetry time-series data which consists of voltage, rotation, pressure, and vibration measurements collected from 100 machines in real time averaged over every hour collected during the year 2015.

2. Errors dataset

The second major data source is the error logs. These are non-critical errors that surface, while the machine is still operational and thus do not constitute as machine failure. The error date and times are rounded to the closest hour since the telemetry data is collected at an hourly rate.

3. Maintenance dataset

These are the scheduled and unscheduled maintenance records which correspond to both regular inspection of components as well as failures. A record is generated if a component is replaced during the scheduled inspection or replaced due to a breakdown. The records that are created due to breakdowns are called failures. The Maintenance data has both 2014 and 2015 records.

4. Failures dataset

These are the records of component replacements due to failures. Each record has a date and time, machine ID, and failed component.

5. Machine dataset

This data set includes some information about the machines like model type and age, the age here is the years in service.

First step would be to save all four data sets into the proper directories on the local machine. We noticed that the final dataset has several unwanted features like Timestamp and maintenance log information. So we decided to exclude those from the DataFrames. Several other columns which had maintenance log information were redundant and were removed. More details about the cleaned dataset can be found here in this link to the Jupyter notebook. The cleaned dataset is now ready for explorations.

3. Data Exploration

3.1. Introduction to the cleaned data

There are 1,021,710 records in the final dataset. There are 4 predictor columns of interest namely the Voltage, Rotation, Pressure and Vibration metrics. The decision column is a description of the component that failed. The target column of interest is the failure column which contains 4 values:

0 for component 1 failure,
1 for component 2 failure,
2 for component 3 failure,
3 for component 4 failure.

After the data exploration we should have clearly defined target columns and feature columns.

Please see the Jupyter Notebook for more details

3.2. Failures

The machine failures were identified, and the components that caused the failure were flagged.

3.3. Other columns

There are columns that identify the non critical maintenance logs. These can be analysed to gather more information on the machine logs as a good project for the future.

4. Model

Knowing the labels for the machine failures, 0 for component 1 failure, 1 for component 2 failure, 2 for component 3 failure and finally 3 for component 4 failure, we use the supervised machine learning algorithms to build a predictive model. Since there are only 4 classes in the data, we use the multi label classification algorithms. The models are trained using 80% of the data and the remaining 20% is used to evaluate the performance of the models.

4.1. Data Pre-processing

Before we start employing different machine learning algorithms, there are some very important routine pre-processing steps that must be performed on the data. We outline these steps below. Even though all steps are NOT required at all times or in other words not recommended for the best results, all the preprocessing steps are listed below. We usually like to keep the order of sequence of steps on the preprocessing steps before running through the classification algorithms.

1. Data Splitting

The first step involves splitting the data into 80% Train set and 20% Test set. Save the training and test datasets into a separate .csv file format.

2. Handle the predictor columns

Imputer:

The imputer will handle any missing (NaN) values in the Predictor columns.

Standardizer:

Use the standardizer to handle the numerical columns in the Predictor columns.

One Hot Encoder(OHE):

Use the OHE to handle the categorical columns in the Predictor columns.

Feature matrix:

The Feature matrix is now ready.

3. Handle the decision columns

Label Encoder:

Treat as categorical columns and use the Label encoder to encode labels. This step will give numerical values to string values.

Target Column:

Processing of the Target column is now complete.

Save the trained models

With the pre-processed predictor columns and decision columns, several classifier models will be considered and all the models will be saved into local pickle files.

4.2. Inverse transform

Inverse transform on the label encoder to get back the true label predictions

APPENDIX