Finding balance between corrective maintenance and preventative maintenance using predictive maintenance

Viral Patel
M. S. Data Science
College of Computer & Information Science
patel.vira@husky.neu.edu

Anant Jain
M. S. Data Science
College of Computer & Information Science
jain.anan@husky.neu.edu

Abstract

Over the last decade radical increase in computational power and the ability to extract meaningful information from large volume of noisy, disparate data at an industry scale has provided many benefits to organizations in all the fields. Predictive maintenance is such field where we can use time series data of machines to provide predictions about its future conditions. In our case we are working with data which is gathered by various IOT devices using sensors to measure the current working condition of the machine. After identifying the target outcome as component failure, we moved to exploratory analysis to identify important features for our dataset. In feature engineering using we have combined different data sources to generate single set of features using one hot encoding. After structuring problem into multiclass classification problem, we have used four supervised machine learning algorithms to classify our data. Using different sets of evaluation techniques, we have compared the results of all the classifiers, from which we have made a conclusion that Boosting Gradient Descent Classifier provides the best results. Random Forest Classifier takes lesser time to train and provides second highest accuracy. Although it may seem GBDT are better learners than random forests, GBDT are prone to overfitting. To overcome this problem and build more generalized trees we have kept learning rate and depth of tree to be on the lower side to allow for better learning.

1. Introduction

The technical meaning of maintenance involves operational and functional checks, servicing, repairing or replacing if necessary devices, equipment, machinery, building infrastructure, and supporting utilities in industrial, business, governmental, and residential installations. Over time, this has come to often include both corrective and preventive maintenance as cost-effective practices to keep equipment ready for operation at the utilization of system life cycle. Corrective maintenance can be defined as maintenance which is carried out after failure detection and is aimed at restoring an asset to a condition in which it can perform its intended function. Preventative maintenance can be described as following planned guidelines from time-to-time to prevent equipment and machinery breakdown using scheduled checks on machine.

Whereas, Predictive Maintenance can be defined as but not limited to predicting possibility of failure of an asset in the near future so that the assets can be monitored to proactively identify failures and take action before the failures occur. These solutions detect failure patterns to determine assets that are at the greatest risk of failure. This early identification of issues helps deploy limited maintenance resources in a more cost-effective way and enhance quality and supply chain processes. Business problems in the predictive maintenance domain range from high operational risk due to unexpected failures and limited insight into the root cause of problems in complex business environments. By utilizing predictive maintenance, we can identify mechanical default detection by their severity and potential impact for disrupting production and operational continuity. By employing such technology in a machine their manufacturer companies can improve overall brand image, eliminate bad publicity and resulting lost sales from customer attrition.

2. Dataset

We combined 5 different datasets which are publicly available on Microsoft Azure cloud.

Machines Dataset contains a set of 1000 machines over the course of a single year (2015). This data set includes information about each machine: Machine ID, model type and age (years in service). The Errors Dataset contains 11967 non-breaking errors recorded while the machine is still operational. These errors are not

considered failures, though they may be predictive of a future failure event. The error datetime field is rounded to the closest hour. The maintenance dataset contains both scheduled and unscheduled maintenance records with a total of 32592 observations. Scheduled maintenance corresponds with regular inspection of components and unscheduled maintenance may arise from mechanical failure or other performance degradations. The telemetry time-series dataset consists of voltage, rotation, pressure, and vibration sensor measurements collected from each machine in real time which contains 8.7 million observations. The data is averaged over an hour and stored in the telemetry logs for over the year 2015. Failures dataset records correspond to component replacements within the maintenance log. Each record contains the Machine ID, component type, and replacement datetime. These records will be used to create the machine learning labels we are trying to predict. It contains 6726 records.

3.Technical Approach

3.1 Exploratory Analysis

We started with exploratory analysis to understand how to interpret the primary patterns in our dataset to understand which features are useful for our model.

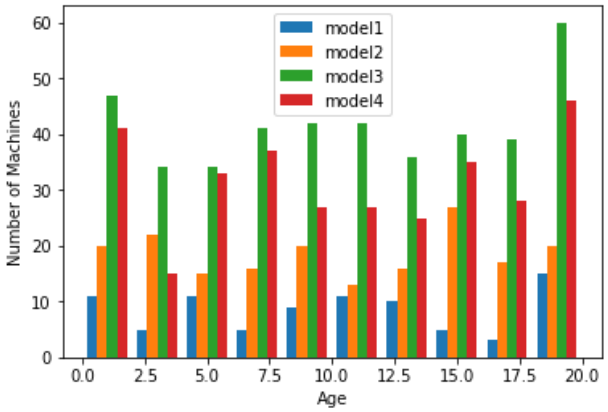


Figure-1: A histogram of the machines' age colored by respective model type

The machine age is an important feature for analysis because various errors and failures of machines largely depend on it. We can see that for all age intervals the number of machines is always highest for model 3 followed by model 4, model 2 and model 1.

The error dataset has datetime data stored and rounded on an hourly rate.

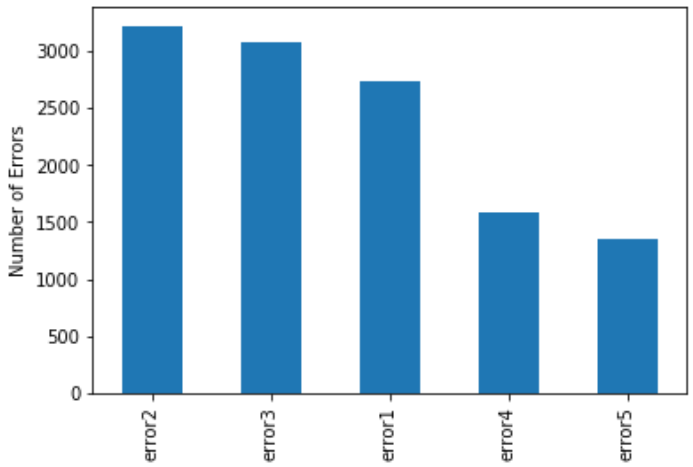


Figure-2: Number of errors for each error type

Here as stated in the introduction about the dataset the error data is a time series data about the five different classes of error which occurred throughout the year. Plotting the error log for each machine will not be much helpful here since there are total of 1000 machines. Thus, it will be very unconstructive to go through analysis of each machine.

As Maintenance records contains both scheduled and unscheduled maintenance checks and a failure record can be generated for both the cases of maintenance. Maintenance records can be used to calculate the component's life, the maintenance data is collected over years 2014 and 2015 instead of only the year of 2015 which is considered in other data.

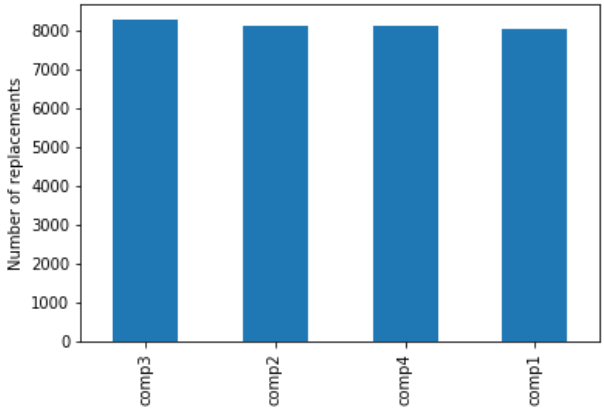


Figure-3: Number of components replaced for each component type

From the graph it can be inferred that the components are replaced at similar rates.

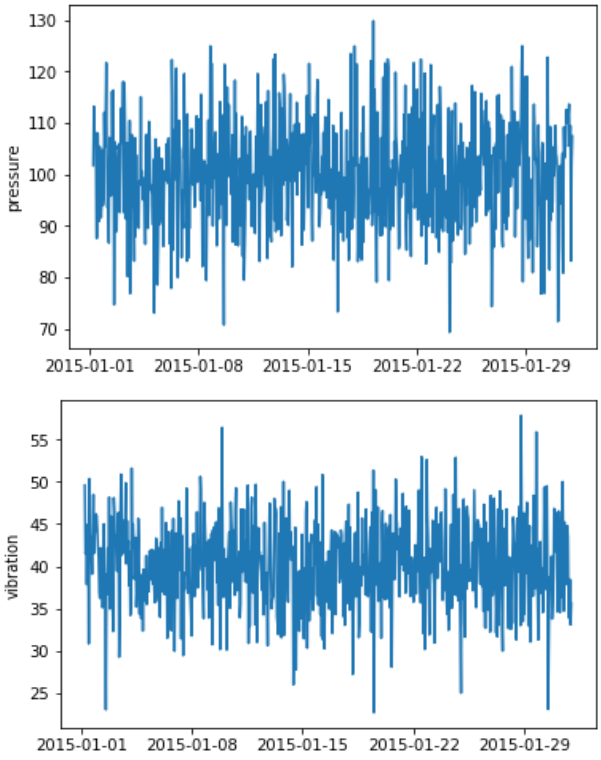
Further analysis on this data can help us understand the underlying pattern about the lifetime of each component or the time history of replacements within each machine.

Preprocessing of this data is discussed in our feature engineering section.

Telemetry is an automated communication system which measures and collects the data using sensors at remote server and transmits it to a recipient equipment for monitoring. Here the it contains real time data of voltage, pressure, vibration and rotation of a machine. The final data is measured by averaging the data over an hour and stored in the telemetry history.

It is important to calculate voltage data since many circuits are designed to handle only certain number of volts. Fluctuation in voltage can also be a reason for circuit failure. Also, vibrational data and rotation data is a useful tool which is heavily used to measure tool's current condition which helps us to calculate remaining life span of that component.

In any heavy machinery industry, operation cycle heavily depends on maintaining right pressure in the machine. For example, during any high-speed operations such as cutting operation it is very important to maintain high pressure of cutting liquid which will help us reduce the wear and tear of the equipment and prevent breakdown of the system.



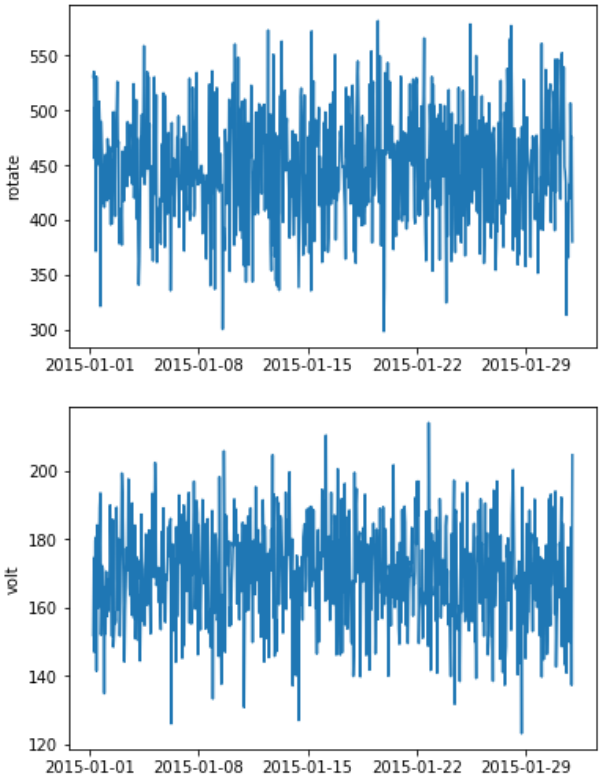


Figure-4: Telemetry data of one machine for one Month, plotted for each type of sensor.

Failures data consists of information about component replacements with their replacement time stamp and machine ID. This dataset is built from component replacement history from maintenance dataset. Using this data, we are creating machine learning labels for predicting failures.

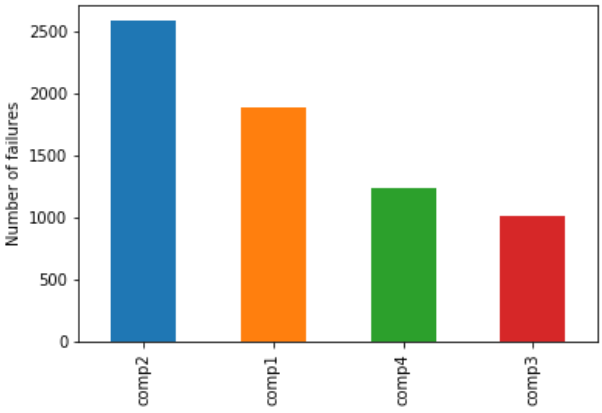


Figure-5: The number of failure related replacements for each component type.

Here we note that component 2 has failed the most number of times.

3.2 Feature Engineering

In this step we have combined different data sources to generate a single data set of features which is used to summarize the machine's life cycle over a period. We have generated a single record for each time unit mentioned in the final dataset. These records contain various features and labels to generate a model which is then provided to the machine learning algorithm to make predictions on the dataset.

A time series data must be restructured as supervised learning problem. There are many different approaches to create features from time series data. we must choose the variable to be predicted and use feature engineering to construct all the inputs that will be used to make predictions for future time steps First and foremost we divide the duration of data collection into time units in which each record points to a single point in time. The measurement unit choice can be considered arbitrary in this case. Time can be measured in months, days, hours, minutes etc. Unit choice is depended on its use case domain.

Furthermore, If the time unit doesn't change frequently we can also use some statistical methods to generalize the pattern of change. For example, if we are calculating temperature change in a day then measurement of temperature every 5 seconds will be trivial for the overall pattern. Instead of that, we can consider the temperature change over a longer time interval which might explain patterns that contribute better for predicting the target variable.

After setting the frequency of observation we are looking for the trends with different measurements over time, to predict performance degradation of machine and connect all this with its probability of failure. Afterwards we are creating features for these trends for each record using time lag window as a parameter over previous observations to determine machine's performance. We use rolling window strategy where we calculate summary statistics such as mean over a window from previous observations.

3.2.1 Telemetry features

From all our datasets telemetry dataset is the largest time series data which has 8761000 hourly observation for 1000 machines. Since, there is not much variation observed for hourly observations, we transform the data by averaging sensor measures over 12 hour rolling window. After this transformation, we are replacing the row data with rolling window data, reducing it to 731000 observations. As a benefit of this approach it has reduced computation time required for feature engineering, modelling and labelling.

After reducing the data, we calculated our lag features by rolling aggregate measures such as mean, standard deviation, minimum, maximum to analyze the intermediate history of telemetry data.

3.2.2 Errors features

As error IDs are categorical values, it can't be averaged over time intervals like above. Instead of that, we are counting the number of errors for each time window.

We transform and align the data by rolling 12-hour window using a join with telemetry data.

3.2.3 Maintenance features

Determining lagging features from the maintenance data isn't an effortless calculation. It requires understanding of various machine domain expertise such as why this component has failed and what parameters can correlate better with its failure pattern.

It is important to consider the number of replacement components and calculate the replacement time of each component. This is because component's failures are highly related to its life cycle. In most cases it is true that the longer the component is used, more prone it is to failure.

We have addressed the problem and calculated the number of replaced components in given time and sorted the data as per recent date of change.

3.2.4 Machine features

Machine dataset includes specification of each machine with its age. These can be used without modification as it includes descriptive information. But as model's description is of string type we created a set of dummy variables to specify the model type.

We merged the maintenance, error, machine and telemetry features into an integrated features dataset.

3.2.5 Label engineering

In a typical classification problem, we just define Boolean labels which can be either true vs false or win vs lose etc. To train the model, the

model requires data from both the classes. In our case, to predict failures the model requires the time series data which leads up to the failures of machines as well as the examples of healthy operational periods of the machine. In our example, the classification is between operational condition of machine which is healthy vs failed.

Once we completed the classification between healthy vs failure, we needed to do some additional calculation. We will only be able to gain advantage of our machine learning algorithm when we can give a notification for the upcoming problem in a machine to prevent the breakdown of a machine. To build a model for that we made modification in the label construction. We changed label definition of *failure event* to a longer window which indicates *failure duration.* Again, the duration of this warning window is highly dependent on the business application. For example, to prevent runway boiler accidents (which occurs when heating unit reaches an exceedingly high temperature and pressure and doesn't shut off after that) the warning window should be kept longer. In such cases, time is a critical factor here which can help us save many lives. It is up to the company to decide whether information about failures will occur in next 24 to 48 hours is enough time to prevent it from happening.

To reconstruct the failure to about to fail we label all observations within failure warning window as failed. Then we estimated the probability of failure within this window.

A categorical feature is created to define as label of failure. For example, All the records which are within 24-hour window before a failure of component 3 are labelled as failure = "comp3", and so on for components 1,2,4. Also the records which are not within seven days of a component failure are labelled as failure = "none".

3.3 Modelling

A general method in machine learning is to train your model with different model parameters and test it on the data which has not been used in training of the model. This kind of evaluation method requires partitioning the data into different sets. Typically, 70% of the data is used for training and 30% is used for testing.

Generally random splitting is used, but here if we used that approach then we will not be able to consider the correlation between these time series observations. For predictive maintenance domain, a time-dependent split is more useful for predicting underlying patterns. For this type of analysis, a single point in time is chosen and then the model is trained on examples up to that point in time and tested on the examples after that point. To account for the failure pattern, we haven't labelled feature records within the split point since it can be categorized as unobserved data.

So, in our analysis we had split data at a single point after 9 months of operation in the year 2015. Below table shows the frequency of each component failure.

3.3.1 Classification Models

Generally, classification is the uneven distribution of data for different classes. In our example, the machine failures are usually rare occurrences compared to instances where machine is categorized in healthy condition. This type of pattern is helpful for businesses, but it can cause imbalance in label distribution. This leads to poor prediction as algorithm tends to classify majority class examples at the expense of minority class. Total misclassification error will be much less because majority of class is labeled correctly which leads to lower precision rate with high accuracy. This type of misclassification can create problems where cost of false alarm is very large. To overcome this problem, sampling techniques such as oversampling of minority examples can be helpful to consider. These methods are not covered in this project because of correlation of data with time. Instead of that we are calculating evaluation metrics for determining precision level along with accuracy.

We had built and compared four different classification models:

Random Forest Classifier: Decision trees are widely used since they are easy to interpret, handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearities and feature interactions. A random forest is an ensemble of decision trees. Random forests combine many decision trees in order to reduce the risk of overfitting. Tree ensemble algorithms such as random forests and boosting are among the top performers for classification and regression tasks.

Naive Bayes Classifier: Naive Bayes Classifier is based on the Bayesian theorem and

is suited when the dimensionality of inputs is very high. In naive Bayes classifier we assume that samples are independent, and all features are given the same weights to classify an outcome. Here each distribution is assumed to have one dimensional distribution which helps in reducing the dimensionality problem. Despite it's simple architecture naive Bayes classifiers works well in real world situations and that is the reason behind choosing this method.

Gradient Boosting Classifier: Like other boosting methods, gradient boosting combines weak "learners" into a single strong learner in an iterative fashion, typically decision trees. GBDT training generally takes longer because of the fact that trees are built sequentially. However, benchmark results have shown GBDT are better learners than Random Forests.

Neural Network Classifier: Neural network algorithm is based on the complex network of neurons in our brain. They process single record at a time and learns from comparing their classification with the actual classification of the record. The errors from the initial classification is then fed back to the network which is used to modify the algorithm for next iterations.

4. Evaluation

To evaluate, we compared the actual failures listed in the test data with the predicted component failures over the test data. We had calculated confusion matrix which lists predicted failure in column and actual component failure in rows.

The diagonal values in confusion matrix shows accurately classified component failures. Numbers above the diagonals indicate inaccurately classified failures when a failure hasn't occurred. Numbers below the diagonal indicates incorrect prediction of non-failure when a failure has occurred in reality.

Accuracy is a measure of how correctly we have predicted the labeled data. But, again when there is a class imbalance this measure becomes biased towards the class with more number of data. Here in our example non-failure class has more number of data then failure class. Due to this we looked at some other statistics as precision, recall and F1 score. positive samples here indicate a failure of a machine. Precision is used to determine how well the model classifies the truly positive samples. It depends on falsely classifying negative days as positive. Recall is used to measure how well the model can find the positive samples. This depends on falsely classifying positive days as negative. F1 is calculated using both precision and recall. F1 score is the harmonic average of precision and recall. F1 score of 1 corresponds to perfect precision and 0 corresponds to the worst performance. The graph below shows important features through which we can make an accurate prediction about future condition of the machine.

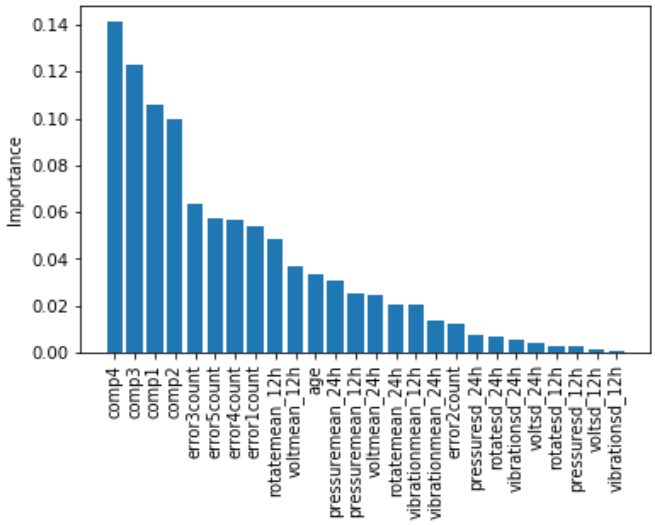


Figure-6: Feature Importance

Model/ Measure	Random Forest Classifier	Naive Bayes Classifierssifier	Gradient Boosting Classifier	Neural Network Classifier
Training Time	2min 18s	951 ms	13min 2s	3min 14s
Accuracy	0.953144	0.876939	0.954191	NA
Precision	0.937656	0.231597	0.874031	NA
Recall	0.368017	0.351589	0.411371	NA
F1	0.528575	0.279248	0.559438	NA

Table-1: Comparison between all four classification models for different measures

From the table we can say that Gradient Boosting has highest F1 score (0.55) and from that we can say that it's the best classifier for our data however it requires more time than other algorithms. Random Forest is also a good classifier with 0.52 F1 score and it also requires significantly less time than Gradient Boosting classifier. Naïve Bayes converges very fast which is below 1 second. Neural Network is not useful with such an imbalanced data and takes very high time with just 10 epochs with very less accuracy. To train such a large-scale data it requires much more time.

Participants Contribution:

Anant Jain:

- Preliminary data clean-up and visualization
- Feature engineering and target labelling
- Model building, comparison and evaluation
- Naïve Bayes, Boosting Gradient Descent and Neural Networks model implementation

Viral Patel:

- Project idea and project goals
- Data tidying and exploratory analysis
- Selection of lag and window features
- Random Forest model implementation
- Documentation

References:

I. https://github.com/Microsoft/SQL-Server-R-Services-Samples/tree/master/PredictiveMaintanenceModelingGuide/Data

II. https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/cortana-analytics-playbook-predictive-maintenance

III. https://en.wikipedia.org/wiki/Maintenance_(technical)#Corrective