# The Application of Predictive Analysis in the Manufacturing Industry

**Michael Hotaling**
Bellevue University, 1000 Galvin Rd S, Bellevue, NE 68005

**Email**:
mjhotaling@my365.bellevue.edu

### Abstract

With the dawn of IoT and Industry 4.0, it is now possible to monitor manufacturing machinery to predict failures before they occur. Most manufacturing systems built today have an array of sensors that report a wide range of variables depending on what kind of work the machinery is designed to do. These sensors can report several key variables that impact production including speed of components, torque measurements, temperature measurements, vacuum levels, laser power and more. All of this can be analyzed on-the-fly in data warehouse systems to detect anomalies and process variations before they have a chance to impact the product.

Keywords: predictive, analysis, maintenance, IoT, industry, machinery

## Background

Since the industrial revolution, machinery has been used to produce a wide variety of goods. The Ford Motor Company is credited with first utilizing assembly lines to streamline the production of the Model T vehicle back in 1913. The assembly line operated on a conveyer system, where the vehicle frame would move along from station to station as workers installed components. This conveyer system allowed Ford to drastically reduce the manufacturing time of vehicles from 12 hours per vehicle to under one hour and 33 minutes (History.com Editors, 2009). This technique allowed Ford to increase production efficiency and lower the cost of their vehicles, making them more accessible to consumers. Between 1914 and 1927, Ford was able to manufacture approximately 10 million Model T vehicles.

In recent years, manufacturing has been dominated by machines. These machines greatly reduce the amount of human error introduced into the assembly and manufacturing process greatly as well as do tasks that are impossible or unsafe for human workers. However, they still at risk for breakdowns and part failure, which can lead to defects in products, longer waiting times for customers, unnecessary waste of raw materials, extra processing time, and in worst case scenarios, total gridlock of the production line; all of which directly impact the revenue from these businesses.

There have been different techniques to combat these failures. In earlier times, it wasn't uncommon to run machinery to failure. This resulted in large amounts of waste from product that ran through the machinery during the mechanical malfunctions, damage to other related components, and caused unscheduled downtime, putting the rest of the manufacturing line at risk. This is known as corrective maintenance, since an issue needed to be corrected before the machinery can return back to production. To avoid unscheduled downtime, preventive maintenance was introduced, allowing engineers to perform checks on a routine basis. This often results in issues being found before they can impact production, but often results in lower production time since machinery needs to be idled down to perform these routine checks and often no issues are found.

## Problem Statement

These two techniques of preventive maintenance and corrective maintenance are both extremes, with one prioritizing uptime while risking the health of machinery and product while the other prioritizes machine health over uptime. With the dawn of IoT and Industry 4.0, it is now possible to monitor manufacturing machinery to predict failures before they occur. Most manufacturing systems built today have an array of sensors that report a wide range of variables depending on what kind of work the machinery is designed to do. These sensors can report several key variables that impact production including speed of components, torque measurements, temperature measurements, vacuum levels, laser power and more. All of this can be analyzed on-the-fly in data warehouse systems to detect anomalies and process variations before they have a chance to impact the product.

Using this data, we can create a predictive model that can send us alarms when our process starts to trend out of spec, preventing any unnecessary unscheduled downtime and loss of material, while also reducing scheduled downtime by isolating issues down to a single source. This will ensure that our machines remain in good health while operating at the maximum possible availability.

## Overview

For this project, I will be exploring process data related to the OCME Vega shrink-wrapper; a machine heavily used in the food and beverage industry to group together loose cans and bottles into packages with plastic shrink film. Kaggle has data available for 519 different observations over the duration of a year, each reporting sensor data for 2048 time-samples for each unit that passes through the machine.

The scope of this project will be to analyse the sensor data to determine the state of the blade mechanism that makes contact with the plastic film to separate the film from a roll. If the machine runs too long without maintenance, the blade will become dull, causing the packaging to fail and release unpackaged material into the rest of the assembly line.

The data in this dataset has no target attributes, so we will need to investigate the application of unsupervised machine learning methods as well as exploring alternatives for identifying unique patterns in each data. This required several hours of analysis and visualization techniques.

The data in this dataset includes the following attributes.

•Timestamp - Timestamp (seconds)
• pCut::Motor Torque - Torque (nM). The force that causes anti-clockwise rotation is considered positive torque and the force that causes clockwise rotation is considered negative torque.
•pCut::CTRL Position controller Lag error - Represents the instantaneous position error between the set-point from the path generator and the real current encoder position of the motor
•pCut::CTRL Position controller Actual position - Cutting blade position (mm)
•pCut::CTRL Position controller Actual speed - Speed of the cutting blade
•pSvolFilm::CTRL Position controller Actual position: Plastic film unwinder position (mm)
•pSvolFilm::CTRL Position controller Actual speed - Speed of the plastic film unwinder
•pSvolFilm::CTRL Position controller Lag error - Represents the instantaneous position error between the set-point from the path generator and the real current encoder position of the motor
•pSpintor::VAX_speed: VAX measurement of performance

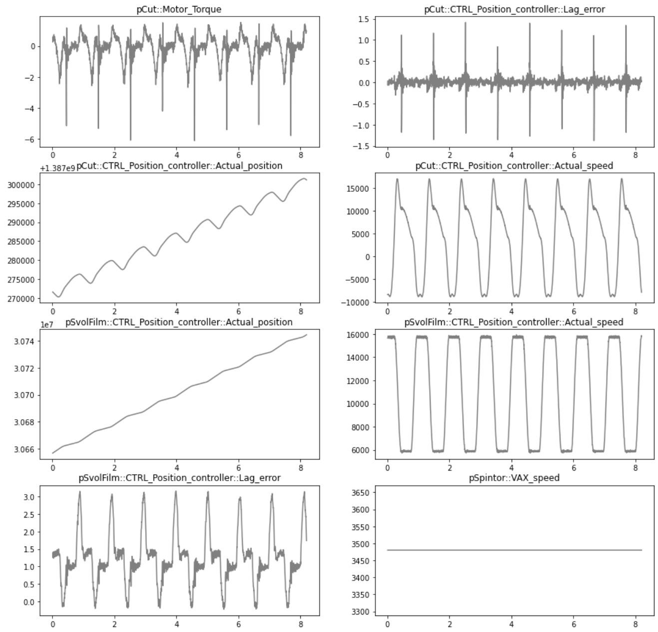


**Figure 1: The OMCE Vega Shrink Wrapper has 8 different sensors, each reporting unique characteristics about the machinery performance.**

Using this sensor data, we can create a predictive model that can send us alarms when our process starts to trend out of spec, preventing any unnecessary unscheduled downtime and loss of material, while also reducing scheduled downtime by isolating issues down to a single source. This will ensure that our machines remain in good health while operating at the maximum possible availability.

## Data Pre-processing and Cleaning

Since this data is comprised of many observations across many different files, some compiling will be necessary to accurately process the data. I will combine all the data into one dataframe and then perform some analysis on that data. Analysis will include generating some summary statistics for each process run, some basic exploratory data analysis (EDA) to find patterns in the data.

The machinery also operates at different speeds depending on the process required. There are 8 modes in total which all have different operating conditions. Therefore, analyzing the difference between these modes and their impact on the system performance should be done.

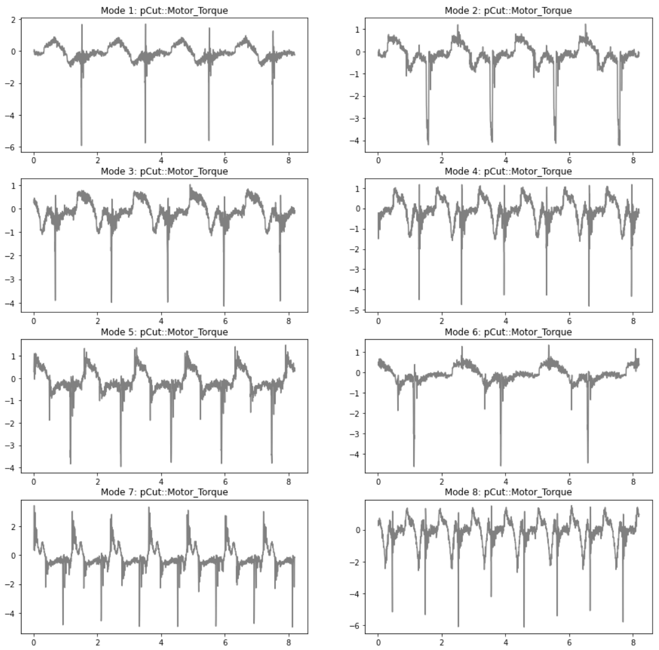


**Figure 2: A visualization of the pCut::Motor_Torque sensor under different operating modes.**

## Identifying Windows

The structure of the data in this dataset is not ideal. For proper implimentation, each packaging cycle should be analyzed. We analyzed the cycles of our sensor data and were able to identify loops using the velocity of certain component in our dataset.

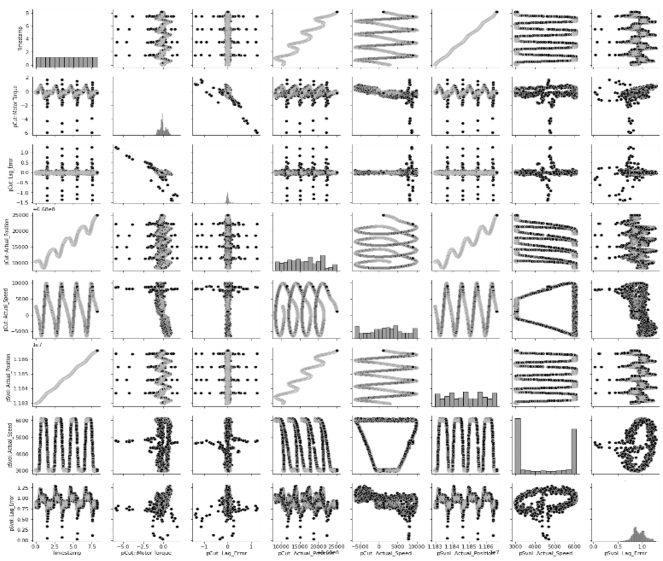

**Figure 3: A pair plot of all of the sensors. Using pCut::Actual_Speed and pSvol::Actual_Speed, we can identify each cycle as the machinery rotates around it's set motions.**

Using these cycles, we can isolate each packaging step within each window of our dataset. *(Kammerer, et al., 2019)*

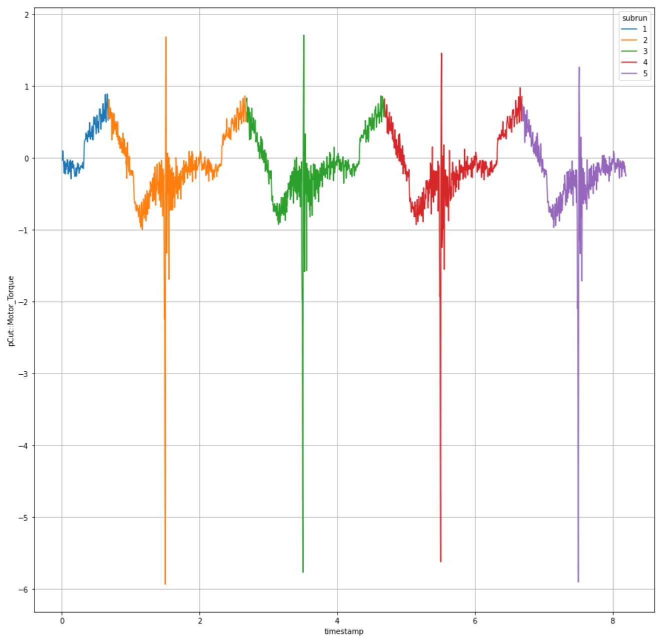


**Figure 4: Segmenting a window from our of our observations by counting the number of cycles.**

Now that each packaging cycle has been isolated in our dataset, we can isolate important sequence in our data. We identiifed 4 unique sequences within each pacakging cycle which we plan to analyze.

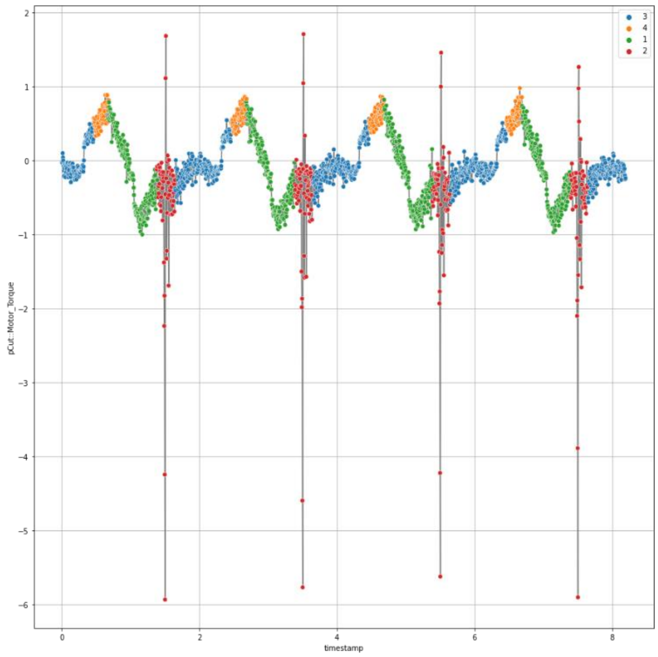

**Figure 5: Identifying sequences of each observation. Our cutting motion occurs in step 2.**

## Generating Summary Statistics

Now that we have successfully identified all of the packaging runs and run sequences, we can calculate descriptive statistics for each packaging run. Key statistics can include the mean, median, minimum and maximum values for the torque, velocity, and lag error sensors for the cutting mechanism.

*(Lewis, 2020)*


**Figure 5: Comparing mean, maximum, and minimum values of pCut::Motor Torque per each cycle. We can see variations in the data as time goes on, indicating some degredation of the hardware.**

We now have several variables in our dataframe, which can be a hinderance in our machine learning modelling. We can reduce the complexity of our dataset by introducing principle components analysis.

## Principal Components Analysis

All of our data points can be represented in n-dimensional space. Principle component analysis reduces the complexity of our dataset by rotating our axes in n-dimesons to fit the data in the best possible method. This is done by calculating the eigenvectors of the covariance matrix of our dataset. The best fit line that explains the majority of our data in our dataset is referred to as the first principal component. The second principal component is vector orthogonal to the first principal component that best describes the perpendicular variance in our dataset.

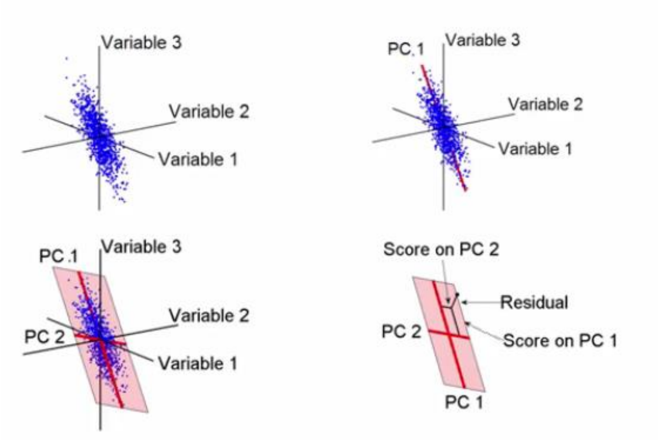


**Figure 6: An example of how principal components analysis works. A line that best fits the variance in the data is drawn through n-dimensions. This represents the first eignevector, or PCA. The second PCA is orthoginal to the first and so on. *(Filion, 2019)***

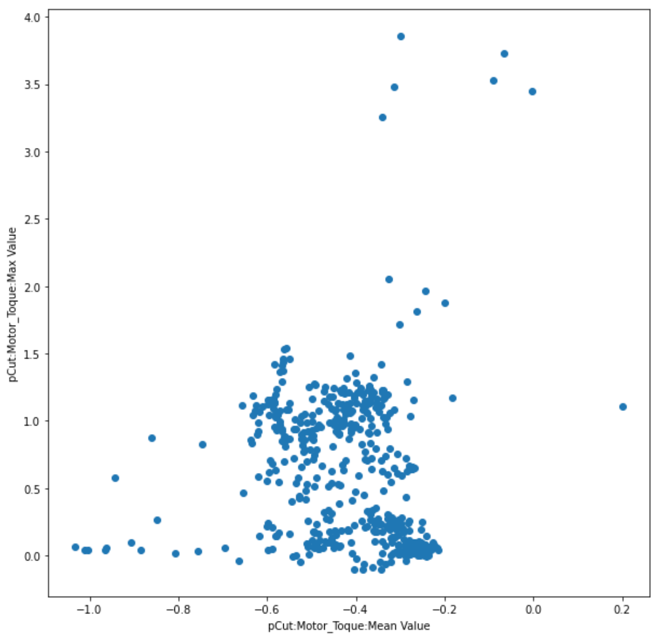


**Figure 7: Comparing pCut:Motor_Torque mean values from step 2 to pCut::Motor_Torque max values from step 2. We have a tight clustering of data points around (-0.4, 0.5) and several outliers surrounding the cluster. *(Filion, 2019)***

By calculating the residuals for each data point based on the scores between all of our principal components, we can establish clusters of data. These clusters should represent healthy operating conditions for our machinery. Any data points operating outside of these expected zones can be considered anomalies and should be documented and inspected.

Running our data through a PCA model, we can see that most of our data is represented within PCA1 and PCA2.

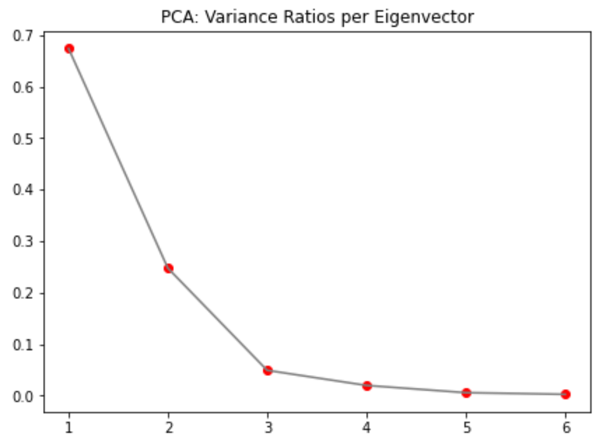


**Figure 8: A plot of the variance rations between each PCA. The first PCA value describes almost 70% of our data. The second describes 25%.**

Plotting PCA1 against PCA2, we can see defined clusters of data. We can assume that these tightly packed clusters are the expected zones of where our machinery should be operating when in healthy condition. We also have a few data points that are outside of that expected zone. These runs can be analysed and compared with the amount of time since the last scheduled maintenance activity. If maintenance hadn't been performed for quite a while, that would validate our hypothesis that the data points trended away from healthy operating conditions. (Lewin & Harmaty, 1994)

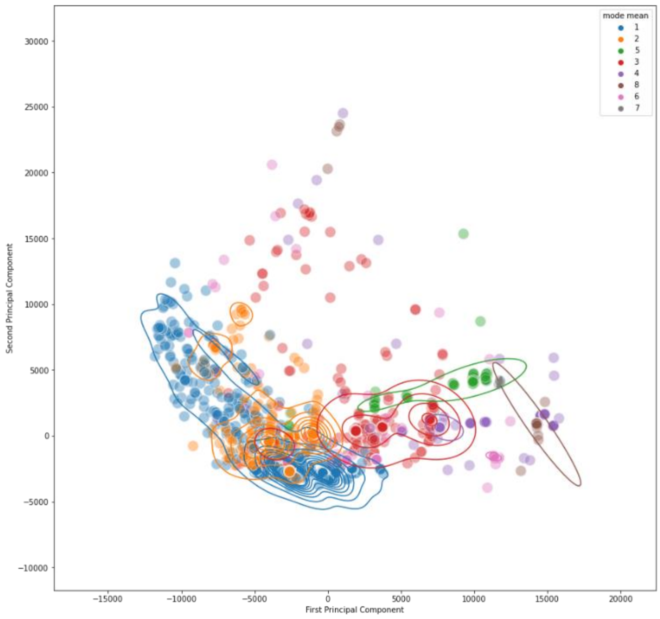


**Figure 8: A scatterplot comparing PCA1 to PCA2. Dense clusters of datapoints can be seen with several datapoints surrounding the cluster. These outliers can be considered anomalies and can be monitored.**

As our machinery degrades away from normal conditions and starts to approach failure, they should start to move outside of the expected zone. This gives us a basic idea of what the data may look like when the machinery degrades away from normal. We can use this to inform our decisions.

## Setting Boundary Conditions

We can set up regions on this graph to determine if the operational statistics can be considered normal or not normal. Our green-zone can be considered normal and won't raise any flags. Our second zone can be labelled as a warning zone. This indicates that our machinery has drifted away from normal conditions, but maintenance isn't required immediately. Very far away from normal conditions, labelled in red, can raise an alarm requiring an engineer to perform maintenance on the machinery or else a failure can be expected soon. (Ran, Zhou, Lin, Wen, & Deng, 2019)

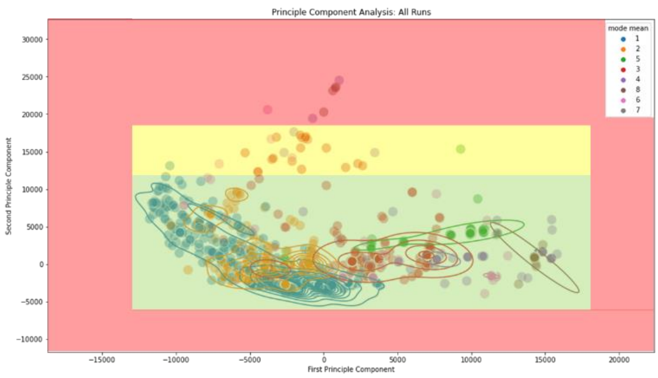


**Figure 8: Defining boundry conditions can help with monitoring the machinery. If our data points drift too far away from the expected normal, our system can raise alerts to notify engineering that the machinery is degrading.**

## Model Deployment

We can collect sensor data on the fly for each one of our packaging cycles. As data comes in, we can generate summary statistics for each cycle that machinery goes through. Then, projecting our data into our calculated principal components, as long as our data.

Since we are working with unsupervised data, validating our model will need to be done by the health of our machinery against any scheduled maintenance tasks. By counting the number of cuts our machinery makes, we can add survival statistics to generate the likelihood of our blade mechanism lasting certain lengths, which should help in model predictions.

## Conclusion

Predictive maintenance is a complex field that is used in all facets of daily life. A common example used would be predicting jet engine health, since running the equipment to failure is not an option and preventive maintenance can cause unnecessary downtime and delays. The PCA modelling methods we have outlined can help reduce the complexity of our data into key attributes and by plotting these attribute against each other, we have demonstrated the ability to visually inspect the machinery's health using only sensor statistics.

Alternations to the model may be needed based on how well the model fits it's needs.

More data would have been helpful for evaluating how well our estimates were. Time between maintenance and number of blade cuts would be key attributes to know as well has having more data in general. Understanding the differences between each operational mode would also be helpful to knowing whether or not these modes have different acceptable thresholds in variance.

## Acknowledgements

We thank Prof. Brett Werner for his help and guidance throughout this project. The completion of this project would have not been possible without his support

## References

Filion, A. (2019, July 2). *Predictive Maintenance with MATLAB*. Retrieved June 5, 2021, from YouTube.com: https://www.youtube.com/watch?v=AS0H43hMoWM

Kammerer, K., Hoppenstedt, B., Pryss, R., Stökler, S., Allgaier, J., & Reichert, M. (2019). Anomaly Detections for Manufacturing Systems Based on Sensor Data -- Insights into Two Challenging Real-World Production Settings. *Sensors*, 1-18.

Lewin, D. R., & Harmaty, Y. (1994). Predictive Maintenance using PCA. *IFAC Symposium on Advanced Control of Chemical Processes*, 25-27.

Lewis, N. (2020, October 8). *Anomaly Detection in Process Control Data with Machine Learning*. Retrieved June 5, 2021, from towards data science: https://towardsdatascience.com/anomaly-detection-in-process-control-data-with-machine-learning-35056a867f5b

Ran, Y., Zhou, X., Lin, P., Wen, Y., & Deng, R. (2019). A Survey of Predictive Maintenance: Systems,. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, 1-36. Retrieved from https://arxiv.org/pdf/1912.07383.pdf