Use of Machine Learning to assess defects in hot rolled coils

Juan G. Sagasti[1], Everett Vazzana[2], Peter Coleman[2], Seth Rummel[2]

[1]AUSTRALTEK LLC
800 Old Pond Rd St 706K, Bridgeville, PA
jsagasti@australtek.com

[2]NORTH AMERICAN STAINLESS
6870 Highway 42 East, Ghent, KY 41045
evazzana@nas.us

Keywords: Hot Rolled Coils, Yield, Defects, Machine Learning, Artificial Intelligence, AI

INTRODUCTION

North American Stainless' Hot Mill is a highly-automated rolling facility with a data collection architecture that stores thousands of data points for each coil produced into their MES databases. A predictive algorithm that calculates the likelihood of certain yield defect before visual inspection takes place was developed using modern Machine Learning (ML) techniques, leveraging the high quality and availability of process data. The implementation process uncovered a number of key facts that helped the company to understand better the nature of the problems.

MOTIVATION

In order to produce high quality products, steel plants implement visual or camera-based quality inspections in key points during the process. The inspection results are used to determine if a product is good or if it must be discarded or reprocessed thus affecting the yield of the line. Since the defects are of heterogeneous variety, the inspectors typically assign a *defect code* to the whole product or to a part of it and that information is used by process engineers to track the source of the problem in order to fix it for future products.

Due to the complexity of the task and the requirement of highly trained personnel, the visual detection procedure is prone to subjective appraisals and misclassifications. This very process of automatically detecting the visual manifestation of defects is being actively researched [1] [2] and there are several competing commercial products that use high definition images and pattern recognition algorithms.

In our case, instead of analyzing the images, the algorithm uses *process knowledge* [3] to pre-qualify the products, providing a numeric value that indicates the likelihood of certain types of yield problems. This information can improve the detection rates by focusing the inspector or inspection system on certain products and patterns.

The pre-qualification score can also be used to perform additional screenings and other measures such as preventive holds. Furthermore, the implementation of such system uncovers a list of process variables that are highly correlated with the yield defects, showing a path for the ultimate goal of fixing them for all future products.

PROJECTS STEPS

In order to obtain a predictive model using ML the following steps were followed:

Dataset Extraction: Build a tool to extract all possible information from several data sources.
Define the Goal: In order to select the best ML model and its parameters we define a metric by which the models will be compared.
Model / Feature Selection: Perform several iterations with different models and different sets of features and compare the selected metric to find a winner.
Evaluation: The results are evaluated in terms of predictive power

MACHINE LEARNING

Machine Learning (ML) is a type of artificial intelligence that provides computers with the ability to predict values without being explicitly programmed, this ability is usually achieved by exposing the algorithm to large amounts of data called *samples*.

ML algorithms can be divided into supervised and unsupervised. A supervised algorithm requires a number of *labeled* samples, the labels being the values or categories that the algorithm is trying to predict on the future *unlabeled* samples. In our case, the historic database of coil inspections can be used as a labeled dataset.

Among the supervised learning methods there are two main categories: Classification and Regression, and classification algorithms can be divided further into binary and multiclass classification.

In this work we will try to detect if an entire coil belongs or not to a category so it constitutes a binary decision, each sample will represent a coil and the label will represent the category of the whole coil.

In our case the label is called *HasDefect* and it can be defined as "Coil has a particular yield defect"

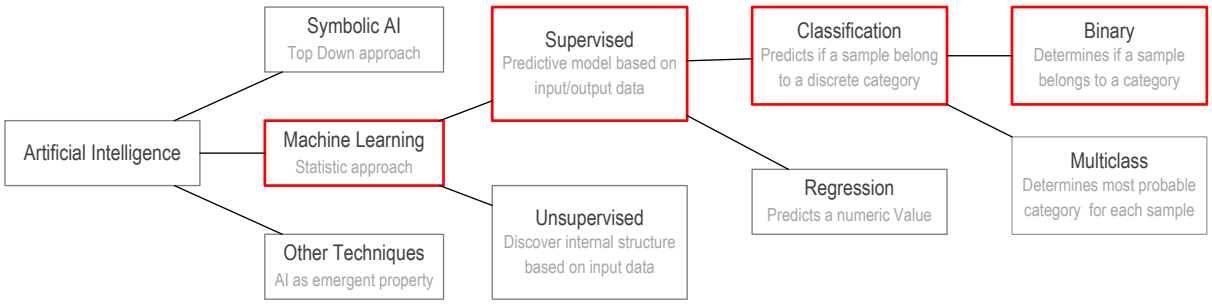

Figure 1 - Different types of Machine Learning algorithms.

DEFINING THE DATASET

We define the *dataset* as the collection of labeled samples. The accuracy of any ML algorithm depends on the quality of the dataset, in our case this means that we need:

- A large number of coils
- A consistent detection of the problem throughout the dataset
- A large and relevant number of process variables assigned to each coil

Features
The measurements / process variables associated with each coil are called *features*. There are two basic types of features that require different treatments: Numeric and Categorical. Examples of numeric features are: Down-coiler Temperature and Average Thickness. Examples of Categorical features are: Steel Grade and Hot Mill Recipe ID.

Missing Values
All the features are acquired using automation systems and are stored in SQL databases. After a close inspection of the data, some coils have NULL values in several features. NULL values cannot be fed to ML algorithms so there are a few options: either remove the sample or provide replacement values.

In some cases, the NULL values represent a valid reason for example "Average Thickness of Pass 6" for coils that only have 5 passes, in those cases we replaced the NULL values by zeros.

In a minority of cases the NULL values were due to other factors and the samples were discarded.

Feature Extraction for trends
The Hot Mill has a large database that contains trends from measurement devices such as thickness gauges and pyrometers, those trends are typically sampled by length or length-adjusted. Each trend can have more than 100,000 data points so is not possible to feed them directly to the algorithm. A usual procedure is to perform Feature Extraction techniques, i.e. calculations based on raw variables. In this case, we divided the coils in three sections: head, body and tail and obtained aggregate calculations such as average, standard deviation, minimum and maximum for each section and for each measurement.

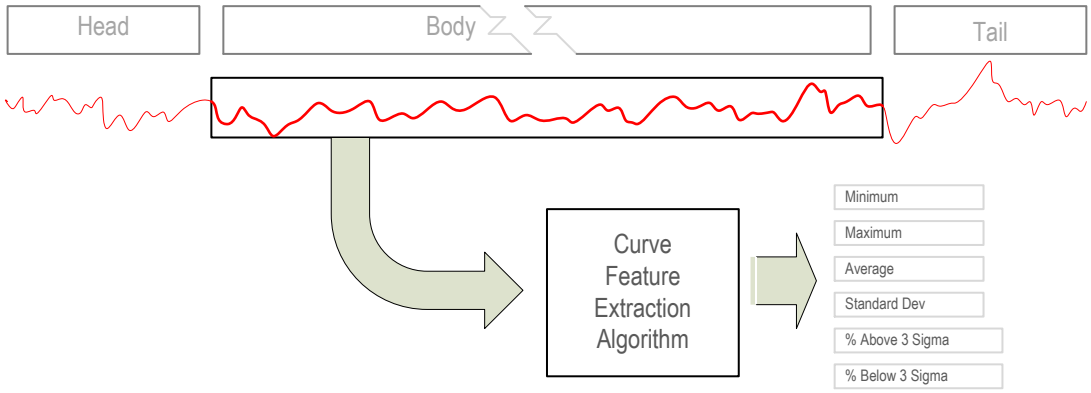

Figure 2- Extracted features for one trend and for the body of a coil

Data Sources
During the model definition, all existing data sources must be considered. In this case, we prepared the data extraction not only to detect this particular yield defect but to be used in the future for other labels or even for performing regression analysis. The following is a list of data sources that we imported into the ML environment.

Slab Chemistry
Slab Dimension and Main Features
Roughing Mill Setup

Roughing Mill Passes
Roughing Mill Trends per Pass
Finishing Mill Setup
Finishing Mill Passes
Finishing Mill Trends per Pass
Down coiler Trends
Crown Measurements per Pass
Inspection Data

The total count of features is around 1600 and the number of coils initially analyzed was above 60,000.

Time Filtered
Due to changes on the way the inspectors qualify this particular yield defect around January 2016, we decided to use only the coils produced after the change, that reduced the number of coils to a figure close to 25,000. Incidentally this operative practice change was independently "discovered" by the algorithm as it was stubbornly detecting *Slab ID (*a simple index number) as an informative feature.

IMBALANCED DATASET

As mentioned above, our label is based on the possession of a particular yield defect. Luckily for the plant the vast majority of coils do not have that defect so our dataset is very imbalanced, meaning that there are much more samples on one category (HasDefect=0) than on the other (HasDefect=1). In ML imbalanced datasets might yield to unstable models or plain wrong predictions. We studied three different approaches to solve this problem

Subsampling: In this approach, we filtered the dataset to obtain all defective coils and a subsample of the good coils, yielding a 50/50 proportion between the two classes. This approach is valid but the downside is that most of the samples end up not being used so their information is discarded.

SMOTE: This data-preparation algorithm creates artificial samples of the underrepresented class (HasDefect=1) increasing the proportion of them to avoid the problem. It works very well in our tests. [4]

Weighted samples: Some models, in particular Logistic Regression, allow the definition of a weight for each sample to penalize errors in the minority class more than the errors on the majority class. We tested this on LR and SVM models with satisfactory results.

DEFINING THE GOAL

When defining a model, one needs to define a scalar numeric goal to maximize or minimize. Classification problems requires a deliberate analysis to find the proper goal because the models can be evaluated using several (sometimes conflicting) metrics.

Confusion Matrix
A good start for understanding the metrics of any binary classification model is the *confusion matrix*

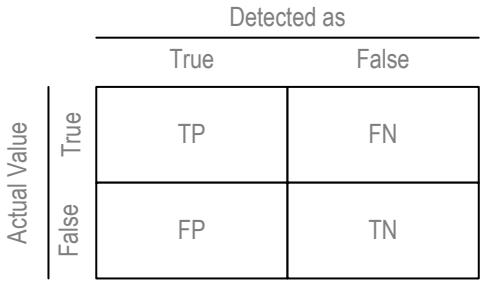

Figure 3 - Confusion Matrix

The four values TP, FN, FP and TN are counts of samples. A perfect model will have FP=FN=0.
In our case:

TP (True Positives): Coils that have the defect and were flagged. AKA Spotted
TN (True Negatives): Coils that do not have the defect and were not flagged AKA Regular Coils
FP (False Positives): Coils that do not have the defect and were flagged. AKA False alarms
FN (False Negatives): Coils that have the defect and were not flagged. AKA Missed.

The following diagram shows a geometric interpretation of a binary classification model reduced to a familiar two-dimensional space. The picture shows the 4 different outcomes for a sample.

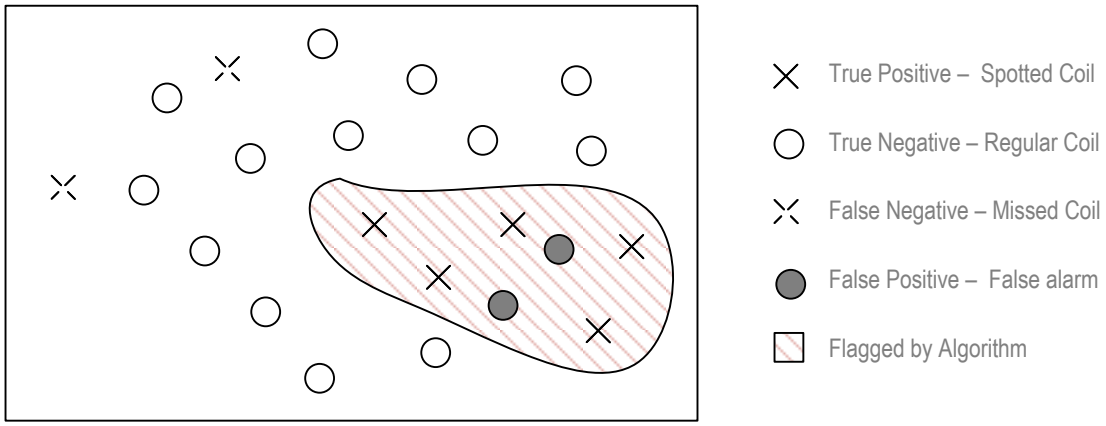

Figure 4 - Geometric interpretation of classification

Binary Classification Metrics
Using the confusion matrix concepts, there are 3 quotients that can be used as metrics:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad , \; Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN}$$

In our case we have to consider two important conditions:

We need a good **recall rate** (i.e. catch the majority of defects) ideally > 90%
We need a good **accuracy rate** (i.e. avoid flagging too many good coils) ideally > 90%

Some models like Logistic Regression and SVM allow selecting a model with a parameter that allows playing with these two metrics until you find a suitable point where both conditions are met. By sweeping that parameter from 0 to 100% a curve called Receiver Operating Characteristic (ROC) is built

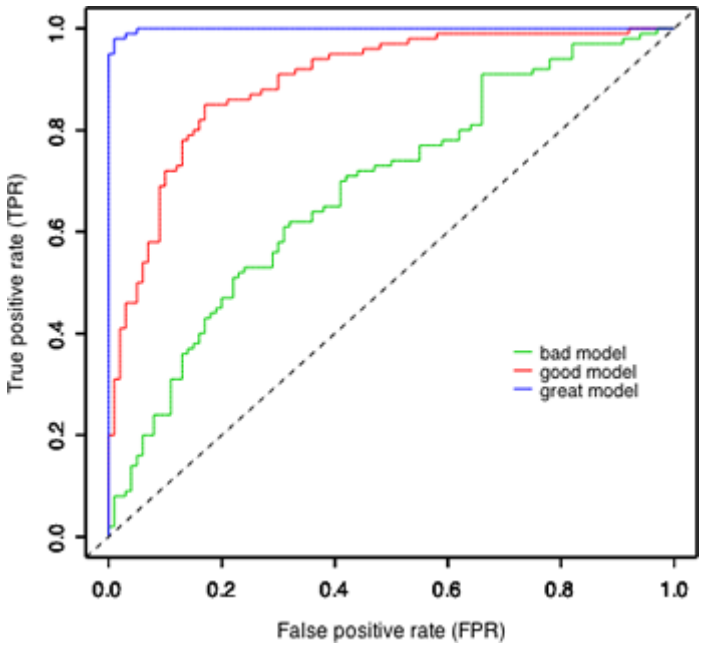

Figure 5 - Receiver Operating Characteristic

We decided to use the area under that curve (ROC-AUC) as our metric to optimize the model, a large AUC implies a model with higher *discriminative power*, the model parameter can then be used to tune the trade-off between acceptable recall and accuracy rates.

IMPLEMENTATION

Software Tools and Algorithms tested

For the development of the models we used a set of tools that included Microsoft Azure ML Studio cloud computing and Python programming language with Pandas and SciKit open source libraries.

We performed several iterations with different models, different parameters and feature sets. Results are shown for Logistic Regression, Support Vector Machines, Boosted Decision Trees and Naïve Bayes.

For each model, we optimized their parameters and used Cross Validation (CV) to assure the models will perform well when exposed to new data points.

Minimizing the feature set

The original dataset has more than 1600 features and it was expected that most of them carry no information about a particular defect. We tried different techniques to select a shorter feature set. Having a short feature set has many implications: improves training speed, improves slightly the AUC metric and more importantly gives process engineers insights of the nature of the defect so they can prevent it in the future.

The method that yielded best results is called “Forward Greedy Selection” [5] and consists in adding one feature at a time, performing several training/scoring loops and finally keeping the feature that maximizes the AUC metric. The process is repeated, adding one feature at each stage until the AUC metric does not increase anymore. The results shown in the following section were obtained with a set of 9 features

BEST RESULTS

Once the feature set was selected, successive tests were running using different models.

In order to account for the imbalance of our set we used SMOTE and found out that 300% as a parameters works out very well

Due to the relative abundance of data we run them, somewhat redundantly, through a 10-fold Cross Validation and score the results with a 20% Test.

For the best performing models we then tuned the parameter (Threshold) to reach acceptable Accuracy and Recall scores.

The following is a summary of the final trials:

Original Dataset				After SMOTE 300%		Selected	Model		Cross Validation 10 fold		Training set 80%			
Rows	Columns	FALSE	POSITIVE	FALSE	POSITIVE	Features	Algorithm	Parameters	Mean	STD	AUC	Threshold	Accuracy	Recall
25644	1606	25424	220	25424	6820	9	TC-BDT	20, 10, 0.2,100	0.995	0.001	935	0.01	932	830
							TC-LR	1E-7,2,2,30	0.945	0.005	949	0.42	922	931
							TCLDSVM	3,1E-1,1E-2,1E-1,1,1500	0.933	0.008	944	0.15	927	931
							TC-SVM	1,1E-2	0.943	0.005	913	0.27	830	931
							TC-Bayes	30	0.944	0.005	948	0.41	910	931

Figure 6 – AUC score comparison between several algorithms and parameters

In this particular case, all algorithms yielded similar results and very strong cross validation scores. The following is the Receiving Operation Curve for the TC-LR algorithm.

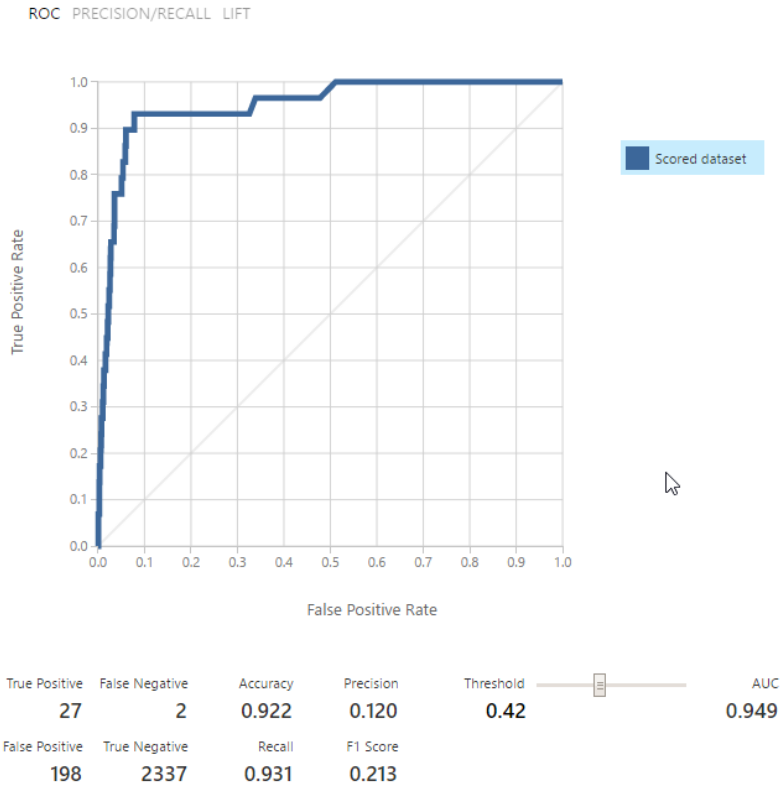

Figure 7 - Evaluation of LR algorithm

This result shows that:

- The algorithm was trained with 80% of the samples and tested with the remaining 20%
- The algorithm cross validation scores indicate **it will perform well on new coils**
- From 29 defective coils the algorithm identified 27 of them: **93% correct**
- From 2535 non-defective coils, the algorithm incorrectly flagged 198: **92% correct**

CONCLUSIONS

The goal of identifying a good predictor function for a particular yield problem was achieved, in the process we developed a data extraction system that can be used for other Machine Learning projects, we implemented a procedure to select a short but performant feature set and discover new ways of exploring causal relationships between defects and process variables.

Regarding the future direction of this study: We are already applying the same data set and concepts to other sets of problems. Another line of action is to take advantage that most of the features are derived from length-adjusted trends so it is possible to increase the granularity of the dataset working with sections of coils instead of complete coils which would yield a better understanding of the physical causes of the labels.

REFERENCES

1. Hongbin Jia ; Y.L. Murphey ; Jinajun Shi ; Tzyy-Shuh Chang: "An intelligent real-time vision system for surface defect detection", Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.
2. Santanu Ghorai ; Anirban Mukherjee ; M. Gangadaran ; Pranab K. Dutta: "Automatic Defect Detection on Hot-Rolled Flat Steel Products", IEEE Transactions on Instrumentation and Measurement, Volume 62 Issue 3. March 2013.
3. Kuldeep Agarwala, Rajiv Shivpuria, Yijun Zhua, Tzyy-Shuh Changb, Howard Huangb: "Process knowledge based multi-class support vector classification (PK-MSVM) approach for surface defects in hot rolling", Expert Systems with Applications Volume 38, Issue 6, June 2011, Pages 7251–7262
4. Nitesh V. Chawla , Kevin W. Bowyer, Lawrence O. Hall W. Philip Kegelmeyer : "SMOTE: Synthetic Minority Over-sampling Technique" Journal of Artificial Intelligence Research 16 (2002) 321–357. June 2002.
5. Shai Shalev-Shwartz, Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms, page 360. Cambridge University Press 2014.