Cement Clinker Free Lime Predictive Model

Paulina González[1], Brenda Rangel[1], Denisse López[2]

[1]Department of Business Informatics Engineering, ITESM

[2]Department of Industrial and Systems Engineering, ITESM

Abstract

Clinker is among the principal components of cement, and the amount of free lime content within this element is considered as the most influential factor to determine the quality of cement. At present, the most common free lime measurement method does not allow for immediate results if the amount of free lime found on clinker is deviant. In practical production, free lime content is mostly measured by offline laboratory analysis, which results in having multiple hour delay results while cement production is a continuous non-stop process. This research aims to use different methods and approaches to build a free lime prediction system that would allow operators to take real-time action. In this study, we use two different data structuring methods and the following model approaches: Logistic Regression, Support Vector Machines and Random Forest Classifier.

1. Introduction

Cement is a construction material with high global demand; reports suggest that 4.1 Gt of cement were produced globally in 2019 [1]. Due to the importance that this construction material has in today's world, cement quality measurement is a crucial aspect of its production. Moreover, cement production is not performed in batches, it is a continuous process that gives no time windows to solve production issues, which escalate even more the importance of its quality measurement methods. This arises a major problem in the cement industry, which is the lack of real-time cement quality measurements. To understand how the quality of cement is measured first we need to briefly describe the production process of cement.

The raw materials that are used for cement production are a mixture of minerals containing calcium oxide (CaO), silicon dioxide (SiO_2), aluminum oxide (Al_2O_3) and ferric oxide (Fe_2O_3). These materials enter a kiln where, at high temperatures, melt and react with each other. This process is called calcination, and it is key to the sintering of the materials and thus the creation of what we know as clinker. Clinker is a semi-finished product and the base component for cement production [2]. The quality of cement is measured by many factors, but the content of free lime (FCaO) in clinker is especially important to judge its quality. Free lime content is the percentage of calcium oxide that does not react with the other components, thereby it is free. The lower the free lime the closer the reactions are to completion. Excess free lime results in undesirable effects such as volume expansion, increased setting time or reduced strength [3]. The free lime target is typically under 1.5% [4].

Offline laboratory analysis is one of the most popular methods of measuring free lime content in cement, and thus controlling its quality. The challenge that this presents is that the results of the free lime content have a significant time delay, which makes it impossible for the operators to detect the problem in real-time, and thus take the necessary measures to fix it. Even though clinker with an atypical content of free lime can be recycled, it represents inefficient usage of resources and increased production costs.

A model for the prediction of free lime content would largely help address these problems, improving production process and thereby resulting in lower revenue loss and more profit. To tackle this problem, we used hardware-sensor data to experiment with three different models: Logistic Regression, Support Vector Machine and Random Forest Classifier to predict free lime classification (optimal quality being below 1.5% of free lime content and low quality being above 1.5%).

2. Dataset, Features, & Evaluation Metrics

2.1 Dataset Overview

The data used for this study was from Cemex, a Mexican multinational company dedicated to the construction industry. The data are specifically from a plant in Brooksville United States and consists of two main datasets: a features dataset and a labels data set. The features dataset consisted of nine hardware-sensor numerical readings which included: preheater temperature (◦F), 4th stage temperature (◦F), calciner temperature 1 (◦F), calciner temperature 2 (◦F), tertiary air temperature (◦F), feeding rate (Tonnage per Hour), kiln motor power (Watts), kiln rotary velocity (RPM), calciner fuel tonnage per hour. Additionally, a binary indicator of the kiln status (on/off) was included. This dataset is a collection of 130,731 readings for each hardware-sensor, with a frequency of these readings being every 15 minutes.

On the other hand, the label's data set consisted of 3,520 readings, having the measurements of calcination (Percentage), free lime content (%), mesh fineness (Microns), and saturation factor of CaO. For this study, we only used the free lime content data, which is measured every hour in an offline laboratory when the kiln is turned on. If the kiln is off, the last lime content measured is repeated in the data set until the kiln is turned on again. The data that was captured when the kiln was off was removed, as it does not represent the normal behavior of the functional oven. In total, there were 112,115 sensor data left, and only 3,381 free lime data. To maximize the usage of sensor data, the approach taken was to use statistical measures that would encompass the behavior of sensor data between each free lime data registration. The following statistical measures were applied and analyzed through statistical and correlational testing: standard deviation, average, Δlast reading - first reading, and Δmax reading - min reading. Only records with time windows where free lime was registered every 1-2 hours were kept (this filter was applied since there were time windows with duration of up to 653 hours). In order to keep the periodicity, the months that had fewer free lime records were also eliminated. The final data set consisted of 2,660 records.

2.2 Data Preprocessing

The first task to properly restructure the data set into time frame windows was to apply statistical measures to the sensor records between each free lime reading. This will encompass all the sensor data that would otherwise be ignored. A student's t-test was then performed between each statistical measure and the free lime content measure for the nine sensors, allowing to know whether the difference between these two groups is statistically significant.

2.3 Student's T-Test

The student's t-test estimates the true difference between two groups' means using the ratio of the difference in group means over the pooled standard error of both groups. It is calculated as follows:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{(s^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right))}}$$

Where t is the t-value, $\bar{X}_1$ and $\bar{X}_2$ are the means of the two groups being compared, s^2 is the pooled standard error of the two groups, and n_1 and n_2 are the number of observations in each of the groups [5]. By applying the student's t-test, we can determine whether two sets of groups are statistically different or not. In this case, being statistically not different indicates the possibility of usage for the indicated statistical measures used to encompass sensor data. Results are shown in Table 1 where "X" indicates the statistical measure for the indicated sensor was not statistically different with the dependent variable.

	1	2	3	4	5	6	7	8	9
Std. Deviation	X								
ΔMax-Min		X					X	X	
Average	X	X	X		X				
ΔLast-First	X		X	X	X	X			

1: kiln motor power | 2 : 4th stage temperature | 3: feeding rate | 4: kiln rotary velocity | 5: calciner fuel tonnage per hour | 6: preheater temperature | 7: calciner temperature 1 | 8: calciner temperature 2 | 9: tertiary air temperature

Table 1: results of student's t-tests, measures statistically not different with dependent variable are marked with "X"

Based on results, the ΔLast-First statistical measure was chosen among the rest for further analysis, since it resulted in the most sensors being statistically not different.

2.4 Feature Analysis

Descriptive statistics were then applied for feature analysis of the already preprocessed data. At first instance, it was determined to ignore the sensor of tertiary air temperature since it is missing large portions of entries and has a large number of outliers. Based on our knowledge of the cement production process we assumed that some features such as the kiln motor velocity and feeding would be highly correlated with the free lime content. Given our initial exploration of the data, it was confirmed that these features are the highest correlated among all others to the free lime content, however, even these correlated features were still only weakly correlated (Figure 1).

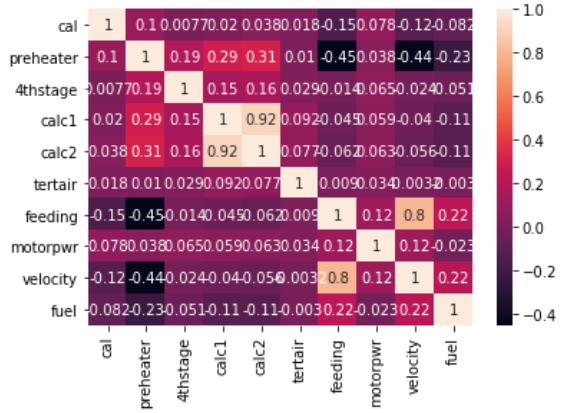

Figure 1: Spearman correlation between variables

Furthermore, we noticed high correlation factors between features that indicated multicollinearity. To begin with the feature selection, we took note of the highest correlated features that would disrupt the model due the redundancy of data. For example, we notice that there is a high correlation between the calciner temperature 1 and calciner temperature 2. From our previous research, we know that there is only one calciner in the process, therefore we can assume there are two sensors measuring the same part of the process. Since both sensors resulted as statistically different from free lime measurements according to the previously made student's t-test, we eliminate both variables from the model.

We also notice a high correlation between the feed rate, kiln's preheater, and motor speed. This can be explained due to the fact that, as the feed rate increases, the kiln's preheater as well as the kiln motor's power must also increase to compensate for the higher feed rate. Since the motor speed and the feed rate have the highest correlation, and the feed rate is more correlated with the dependent variable, we proceed by ruling out the motor speed sensor for our model.

In a further analysis to detect any non-linear correlations, we found no evidence that any other types of correlations exist with the free lime or among the features. The scatter plots showed worrying information; there was no clear distinction between free lime content that derived from either high or low temperatures. The distribution of sensors in both classifications of the dependent variable also behaved in similar ways. In other words, a high free lime content was present in both

high and low temperature and vice versa. To confirm this, a Principal Component Analysis (PCA) was implemented to check if this method could make a clearer distinction between the high free lime content and the low free lime content by reducing its dimensions. Given m dimensions in this case sensors and n principle components, PCA maps data from R_m to the subspace $S \subset R_n$ that preserves most of the variance in the data. Unfortunately, the same results were given, showing no clear distinction between free lime content (Figure 2).

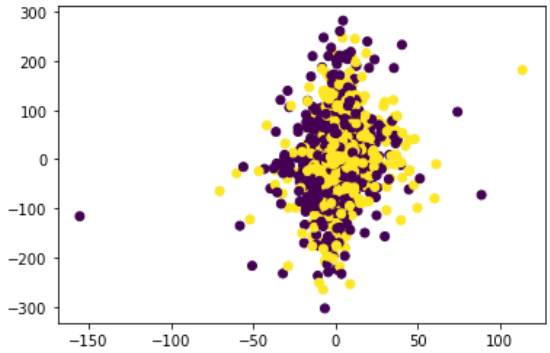

Figure 2: Principal Analysis Component (PCA)

Going on from these conclusions we then used the wrapper method of backward elimination and a VIF analysis to select the best subset of features, these are preheater temperature, feeding rate, kiln motor power and calciner fuel tonnage per hour.

2.5 Synthetic Minority Oversampling Technique (SMOTE)

The final dataset used for model creation was highly unbalanced and therefore needed to undergo a data augmentation technique. One of the main solutions to adjust the balance for biased data is the synthetic minority oversampling technique (SMOTE) [6]. The core idea is that the artificial instance for minority instances is generated using k-nearest neighbors of sample. In the minority instance, k-nearest samples are selected from sample *X*. Afterward, the SMOTE algorithm selects n samples randomly and save them as *Xi*. Lastly, the new sample *X´* is generated based on the below equation.

$$X´ = X + rand \times (Xi - X),\ i = 1, 2, ..., n$$

where *rand* follows a random number uniformly distributed in the range (0, 1). By obtaining minority instances using SMOTE, the class imbalance is reduced thus allowing machine learning and deep learning algorithms to learn better [7].

2.6 Evaluation Metrics

Fundamentally, the goal of our predictor is to anticipate if the clinker will have an optimal or low quality, according to the free lime content in it (optimal quality being below 1.5% of free lime content classified as 0 and low quality being above 1.5% classified as 1). Hence, this will be treated as a binomial classification method. If the model effectively predicts low quality clinker, the operators will be able to act and adjust parameters beforehand. On the other hand, if the model predicts optimal quality clinker accordingly, operators will keep the current process' parameters, avoiding to make adjustments (by altering volume, heat, etc.) and thus negatively affecting the clinker's quality (and possibly increasing costs). Therefore, we can conclude that both classes are as important to correctly predict in this study.

Since the dataset has been balanced through SMOTE data augmentation technique and there is no more importance towards true positives or true negatives prediction, accuracy and AUC evaluation metrics will be used. Predictive accuracy is the performance measure generally associated with machine learning algorithms, which represents the number of correctly classified data instances over the total number of data instances and it's defined as follows:

$$\frac{TP + TN}{TP + FP + TN + FN}$$

where *TN* is the number of negative examples correctly classified (True Negatives), *FP* is the number of negative examples incorrectly classified as positive (False Positives), *FN* is the number of positive examples incorrectly classified as negative (False Negatives) and *TP* is the number of positive examples correctly classified (True Positives) [8].

To complement this metric, we are also using AUC ROC as an additional evaluation metric. The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the True Positive Rate against the False Positive Rate at various threshold values and essentially separates the 'signal' from the 'noise'. The Area Under the Curve

(AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. It is interpreted as being the average value of sensitivity for all possible values of specificity [9].

3. Methods

3.1 Logistic Regression

In logistic regression, a categorical dependent variable *Y* having *G* (usually *G = 2*) unique values is regressed on a set of p independent variables $\mathcal{X}_1$, $\mathcal{X}_2$, ..., $\mathcal{X}_p$ [10].

The logistic function is of the form:

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$

where μ is a location parameter (i.e. the mean) and s is a scale parameter proportional to the variance.

We decided to use logistic regression since it is a simple and easy to interpret method for binary classification models [11].

3.2 Support Vector Machine

Support Vector Machines, also known as SVMs, are known to produce equally good, if not better results than neural networks, while being more efficient and producing an actual mathematical function [12]. For these reasons, and since SVMs have been proved to have good performance when trained with small datasets, we decided to move forward with this model as it could provide better results than the logistic regression.

A support vector's machine goal is to find a function f(x) that has at most ε deviation from the obtained targets y_i for all the training data. The tacit assumption is that such function *f* actually exists that approximates all pairs (x_i , y_i) with ε precision. Sometimes, however, this may not be the case. Hence, it analogously utilizes a "soft margin" loss function that introduces the slack variables ξi, ξi∗ to cope with the otherwise unfeasible constraints of the optimization problem which is defined as:

$$minimize \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i, \xi_i^*)$$

$$subject\ to \begin{cases} y_i - \langle w, x_i\rangle - b \le \varepsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

SVMs also allow us to assume that f(x) is non-linear, if so, the data can be mapped into a higher dimensional space, called kernel space. This replaces all instances of x with k(x_i, x_j), transforming it from feature to kernel space. In this case, Radial Basis Function kernel (RBF) also called Gaussian kernel proved to work best, which is defined as:

$$K_{RBF}(x, x') = \exp\left[-\gamma\|x - x'\|^2\right]$$

where γ is a parameter that sets the "spread" of the kernel [13].

3.3 Random Forest Classifier

Random forest distinguishes from our previous models given that it is an ensemble method, meaning that it is made up of a large number of decision trees called estimators, which each produce their own predictions. The random forest model combines the predictions of the estimators to produce a more accurate prediction. Since it also helps to avoid overfitting, this algorithm was also tested for our model [14].

4. Results, Discussion & Future Work

All results come from the optimized versions of the models discussed. This was done by fine-tuning the hyperparameters of each model utilizing Sckit-Learn's GridSearchSV with cross-validation to evaluate all possible combinations of hyperparameter values and finding the best combination.

4.1 Results

Model	Set	Accuracy	AUC ROC
Logistic	Train	0.5608	0.5590
Logistic	Test	0.5320	0.5289
SVM	Train	0.5752	0.5638
SVM	Test	0.5658	0.5572
Random Forest	Train	0.5897	0.5762
Random Forest	Test	0.6034	0.5897

Table 2: Comparison table of all models' performances

All tested models performed among the same accuracy and AUC ROC range on both the train and test set. The best result was achieved by the Random Forest Classifier, with an Accuracy of 60.34% and AUC ROC of 58.97% on the test set.

This accuracy as well as AUC ROC results are low, showing that the model is underfitting (has high bias). After a thorough analysis of the models and their corresponding training, we concluded that the models were not underfitting due to some training mistake, poor future engineering/selection, or any under adjustments of hyperparameters. The previous descriptive analysis created makes a strong case that the models are having low performance due to the quality and quantity of the data. The extremely low levels of correlation between the features and the target variable were unusual (Figure 1). Additionally, there was never a clear distinction between high and low content of free lime, even when utilizing PCA (Figure 2). Furthermore, high levels of contamination were observed in the data cleaning process leaving a small dataset behind to work with.

4.2 Discussion

There are inherent limitations and challenges when training machine learning models with small datasets. In this study, the response variable was limited; it can be assumed that the performance of the model could be significantly improved with more data. On the other hand, the data showed an abnormal behavior on the free lime content, showing no clear distinction in high free lime sensors from those of low free lime making it almost impossible to obtain an accurate model. This could be due to the fact that workers manually record the free lime content, which implies that the values could have been rounded or not registered accurately. Another explanation is that the sensors are not calibrated well enough or even damaged, thus causing bad readings. Lastly, the kiln indicator in the dataset stated large gaps of "off" status, meaning loss of independent variable data.

4.3 Future Work

Another algorithm that can be implemented with the current dataset is XGBoost, which is an ensemble method similar to Random Forest, but that has proven to be more efficient with small datasets. In the future, it is expected to have a larger and better-quality dataset that can demonstrate the theoretical behavior of the free lime content with respect to the sensors that are involved in the process. Multilayer Perceptron neural network (MLP) implementation can be an alternate model that can adapt better and significantly increase the models' performance if a larger data set is given. Another alternative is to use the dataset of another plant that has more periods of on status in the kiln.

Reference

[1] P. Levi, T. Vass, H. Mandova, and A. Gouy, "Ce- ment," International Energy Agency, Tech. Rep., 2020.

[2] H. Zhang, Building Materials in Civil Engineering. Oxford (GB): Woodhead Publishing, 2011, ch. Cemen

[3] Thermo Scientific. (n.d.). *Free Lime Determination in Clinker ARL 9900 Series with IntelliPower™ Simultaneous-Sequential XRF Spectrometer*.

[4] N.-O. E. Moses and S. B. Alabi, "Developments in the measurement and estimation methods for cement clinker quality parameters," International Journal of Engineering Research & Technology, vol. 5, no. 5, 201

[5] R. Bevans. (2020). *An introduction to T-Tests: Definitions, formula and examples*. Scribbr.

[6]N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res., 16(1), pp. 321-357. (2002)

[7] Y.Won, J.Dirmanto & B.Shivam. *Push For More: On Comparison of Data Augmentation and SMOTE With Optimised Deep Learning Architecture For Side-Channel.* Temasek Laboratories at Nanyang Technological University, Singapore, pp. 3-6. (2020)

[8] N.V. Chawla, K.W. Bowyer, L.O. Hall & W. P. Kegelmeyer. *SMOTE: Synthetic Minority Over-sampling Technique.* Journal of Artificial Intelligence Research 16 (2002) 321–357

[9] Zhou, X. H., Obuchowski, N. A., & McClish, D. K. (2011). *Statistical methods in diagnostic medicine*. Wiley-Blackwell.

[10] *Logistic regression - ncss-wpengine.netdna-ssl.com*. (n.d.). Retrieved February 4, 2022, from https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Logistic_Regression-Old_Version.pdf

[11] Moore, D. S., McCabe, G. P., & Craig, B. A. (2017). *Introduction to the practice of Statistics*. W.H. Freeman, Macmillan Learning.

[12] S. M. Clarke, J. H. Griebsch, and T. W. Simpson, "Analysis of support vector regression for approx- imation of complex engineering analyses," Journal of Mechanical Design, pp. 1077–1087, 2004.

[13] A. J. Smola and B. Scho¨lkopf, "A tutorial on support vector regression," Statistics and Computing 14, p. 199–222, 2004.

[14] G. Biau, "Analysis of a random forests model," Journal of Machine Learning Research, vol. 13, pp. 1063–1095, 2012.