Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Department of Mathematics and Physics

Master of Data Science

Discharge Moisture Prediction of the Corn Gluten Feed Drying Process Using Machine Learning Algorithms

---

THESIS to obtain the **DEGREE** of
MASTER OF DATA SCIENCE

A thesis presented by:
Adrián Josué Garay Gutiérrez

Thesis Advisors:
Rocío Carrasco Navarro

Tlaquepaque, Jalisco, November, 2022

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Maestría en Ciencia de Datos

Predicción de Humedad de Descarga del Proceso de Secado de Gluten de Maíz Usando Algoritmos de Aprendizaje Automático

---

TESIS que para obtener el **GRADO** de
MAESTRO EN CIENCIA DE DATOS

Presenta:
Adrián Josué Garay Gutiérrez

Asesora:
Rocío Carrasco Navarro

Tlaquepaque, Jalisco, Noviembre, 2022

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Department of Mathematics and Physics

Master of Data Science Approval Form

Thesis Title: **Discharge Moisture Prediction of the Corn Gluten Feed Drying Process Using Machine Learning Algorithms**

Author: **Adrián Josué Garay Gutiérrez**

Thesis Approved to complete all degree requirements for the Master of Science Degree in Data Science.

Thesis Advisor, **Rocío Carrasco Navarro**

Thesis Reader, **Juan Diego Sánchez Torres**

Thesis Reader, **Riemann Ruiz Cruz**

Academic Advisor, **Rocío Carrasco Navarro**

Tlaquepaque, Jalisco, November, 2022

Discharge Moisture Prediction of the Corn Gluten Feed Drying Process Using Machine Learning Algorithms

Adrián Josué Garay Gutiérrez

Abstract

Modern manufacturing processes have multiple sensors (or instruments) installed that provide constant data stream outputs; however, there are some critical performance and quality variables where installing physical sensors is either impractical, expensive, not hardy enough for hostile environments or the sensor technology is not sufficiently advanced. An example of such a problem is measure moisture of solid products in real-time. If this scenario happens, Machine Learning (ML) approaches are a suitable solution as are capable of learning and representing complex relationships. ML algorithms establish a mathematical relationship between the quantity of interest and other measurable quantities such as readings from already available sensors (e.g., SCADA, historian softwares, SQL Databases, etc.). This study details how ML algorithms (Such as Multiple Linear Regression, Support Vector Machine Regression and Regression Trees) are used to predict critical variable **moisture** in gluten feed (a by-product of the wet-milling of maize grain for starch or ethanol production) as a simple, robust and fast solution for the lack of this variable real-time information for a corn products manufacturer. The resulting model performance demonstrates the feasibility of the ML algorithms approach to predict **moisture** behaviour.

KEYWORDS: Machine Learning, Virtual Sensors, Multiple Linear Regression, Support Vector Machine Regression, Regression Trees, Drying Process.

Predicción de Humedad de Descarga del Proceso de Secado de Gluten de Maíz Usando Algoritmos de Aprendizaje Automático

Adrián Josué Garay Gutiérrez

Resumen

Los procesos de manufactura modernos tienen múltiples sensores (o instrumentos) instalados que proporcionan flujos de datos constantes; sin embargo, existen algunas variables críticas de rendimiento y calidad en las que la instalación de sensores físicos es o poco práctica, o costosa, o no es resistente para entornos hostiles o la tecnología del sensor no es lo suficientemente avanzada. Un ejemplo de tal problema es medir la humedad de productos sólidos en tiempo real. Si nos encontramos ante este escenario, soluciones con enfoque de aprendizaje automático (ML por sus siglas en inglés) son adecuadas, ya que son capaces de aprender y representar relaciones complejas. Los algoritmos de aprendizaje automático establecen una relación matemática entre la variable de interés y otras variales medibles, tales como lecturas de sensores ya disponibles (p. ej., SCADA, softwares de historización, bases de datos SQL, etc.). Este estudio detalla cómo los algoritmos de aprendizaje automático (tales como regresión lineal múltiple, regresión de máquinas de vectores soporte y árboles de regresión) se utilizan para predecir la variable crítica **humedad** en el gluten de maíz (un subproducto de la molienda húmeda del grano de maíz para obtener almidón o etanol como producto final) como una solución simple, robusta y rápida a la falta de información en tiempo real de esta variable para un fabricante de productos de maíz. El desempeño del modelo resultante demuestra la viabilidad del enfoque de algoritmos de aprendizaje automático para predecir el comportamiento de la **humedad**.

Contents

Page

1 Introduction 17
1.1 Background 18
1.2 Justification 20
1.3 Problem Statement 20
1.4 Aim and Objectives 20
1.4.1 Specific Objectives 20
2 State of the Art 23
2.1 Virtual Sensors 23
2.2 Virtual Sensors in industrial applications 24
3 Theoretical Framework 27
3.1 Product and Process Description 27
3.1.1 Corn Gluten Feed 27
3.1.2 Wet Milling Process 28
3.1.3 Drying unit operation 28
3.1.4 Corn Gluten Dewatering and Drying 31
3.2 Machine Learning 31
3.2.1 Data Preprocessing 32
3.3 Feature Selection Procedures 35
3.3.1 Stepwise Regression 36
3.3.2 Backward Elimination 38
3.3.3 Principal Component Analysis (PCA) 38
3.3.4 Partial Least Squares Regression (PLS) 40
3.4 Linear Regression 41
3.4.1 Ordinary Least Squares Method 42
3.5 Support Vector Machines (SVM) 44
3.6 Decision Trees Regression 46
3.7 Bagged Tree Algorithm 48
3.8 Random Forest Tree Algorithm 48
4 Methodology 51
4.1 Data Description 52
4.1.1 Attribute Information 53
4.2 Data Preprocessing 53
4.2.1 Data Profiling 53
4.2.2 Data Cleansing 54

4.2.3 Data Reduction 55
4.2.4 Data Transformation 61
4.2.5 Data Validation 62
4.3 Linear Regression Application 63
4.3.1 All Data Linear Regression Model (ADLRM) 64
4.3.2 All Features with Raw Data Linear Regression Model (AFRLRM) 66
4.3.3 All Features with Standardized Data Linear Regression Model (AFSLRM) 66
4.3.4 Backward Elimination with Standardized Data Linear Regression Model (BESLRM) . . 66
4.3.5 Stepwise Regression with Standardized Data Linear Regression Model (SRSLRM) 68
4.3.6 Operational Features with Standardized Data Linear Regression Model (OFSLRM) 68
4.3.7 Principal Component Linear Regression Model (PCALRM) 68
4.3.8 Partial Least Squares Regression Model (PLSLRM) 70
4.4 Support Vector Machine (SVM) Regression Application 70
4.4.1 All Features SVM Regression Model: Linear kernel (AFLSVM) 71
4.4.2 All Features SVM Regression Model: RBF kernel (AFRSVM) 71
4.4.3 Feature Reduction SVM Regression Model: Linear kernel (FRLSVM) 71
4.4.4 Feature Reduction SVM Regression Model: RBF kernel (FRRSVM) 71
4.5 Regression Trees Application 71
4.5.1 Decision Tree without Max Depth Defined (DNDRTM) 72
4.5.2 Decision Tree with Best Depth as Hyperparameter (DDHRTM) 72
4.5.3 Bagged Trees (BGGRTM) 72
4.5.4 Bagged Trees with Best Trees Number as Hyperparameter (BTHRTM) 73
4.5.5 Random Forest Trees (RFRRTM) 73
4.5.6 Random Forest Trees with Best Trees Number as Hyperparameter (RTHRTM) 74
5 Results and Discussion 75
5.1 Results 75
5.1.1 Multiple Linear Regression Models Comparative 76

5.1.2 Support Vector Machine Regression Models Comparative 77
5.1.3 Regression Trees Models Comparative 78
5.2 Discussion 79
6 Conclusions. 81
6.1 Conclusions 81
6.2 Future Work 81
Bibliography 83
Index. 87

List of Figures

Page

1.1 Moisture comparison models. 19
1.2 Moisture comparison samples. 19
1.3 NIR online instrument example 20

2.1 The Virtual Sensor Concept. 24

3.1 Corn gluten feed applications 27
3.2 Corn composition 28
3.3 Wet milling process block diagram 29
3.4 Drum Dryer typical schema 30
3.5 Rotary Drum Filter 31
3.6 A diagram depicting the structure of a PLS model. PLS finds components that simultaneously summarize variation of the predictors while being optimally correlated with the outcome 41
3.7 Example of a regression tree model 47

4.1 Gluten dryer real-time data from process sensors. 52
4.2 Missing Data Heatmap for Gluten Drying Process Dataset. 54
4.3 Correlation Heatmap for gluten drying process subset. . 56
4.4 Correlation between GF & TI. 60
4.5 Missing Data Heatmap for Clean Gluten Drying Process Dataset. 60
4.6 Correlation Heatmap for clean Gluten Drying Process Dataset. 61
4.7 Influence plot for Gluten Drying Process dataset. 65
4.8 All Data Linear Regression Model residuals histogram . 65
4.9 All Data Linear Regression Model residuals qqplot . . . 67
4.10 Cumulative explained variance screeplot. 69
4.11 RMSE VS. Number of PC plot. 69
4.12 MSE VS. Number of PC regressors 70
4.13 R^2 score VS. Tree max depth 72
4.14 Score difference VS. Tree max depth. 73
4.15 R^2 score VS. Trees number 73

4.16 R^2 score VS. Trees number 74

5.1 SRSLRM trend plot. 77
5.2 FRRSVM trend plot. 78
5.3 BGGRTM trend plot. 79
5.4 Block diagram of moisture visualization solution. 80

List of Tables

Page

4.1 Target information, such as feature description, scale and unit of measure . . . 53
4.2 Attributes information, such as feature description, scale and unit of measure . . . 57
4.3 Data quality report for Gluten Drying Process Dataset. . 58
4.4 Data quality report for Gluten Drying Process subset . . 59
4.5 Data quality report for clean Gluten Drying Process . . . 60
4.6 Cause of skewness of each variable . . . 62
4.7 Data quality report for train "Raw" Gluten Drying Process subset . . . 63
4.8 Data quality report for test "Raw" Gluten Drying Process subset . . . 64
4.9 Data quality report for train standardized Gluten Drying Process subset . . . 66
4.10 Data quality report for test standardized Gluten Drying Process subset . . . 67
4.11 Features obtained by backward elimination method . . . 68
4.12 Features obtained by stepwise regression method . . . 68

5.1 Results of Multiple linear regression models . . . 76
5.2 Results of Support Vector Machine regression models . . 77
5.3 Results of Regression Trees models . . . 78
5.4 Results of selected models of each scheme. . . . 79

Dedicated to my Company on sleepless nights; to Music that calms my worst frustration and motivates me to push it to the limit to finish this race. Thank you for being. To ALMEX staff, Andrea Fernández, Juan Pablo Pimienta and Salvador Estrada, for allowing me to carry out this work and for human quality shown. To my teachers and classmates at ITESO for resist my endless talks about corn business. To my family and, particularly, to my mother Rosa Ma. Gutiérrez.

Dedicada a mi Compañía en las noches de desvelo; a la Música que calma mi peor frustración y que me motiva a esforzarme al máximo para terminar esta carrera. Gracias por ser y estar. Al personal de ALMEX, Andrea Fernández, Juan Pablo Pimienta y Salvador Estrada por permitirme llevar a cabo este trabajo y por la calidad humana mostrada. A mis maestros y compañeros del ITESO por resistir mis pláticas interminables del negocio del maíz. A mi familia y, particularmente, a mi madre Rosa Ma. Gutierréz.

1 Introduction

Contents

1.1 Background 18
1.2 Justification 20
1.3 Problem Statement 20
1.4 Aim and Objectives 20
1.4.1 Specific Objectives 20

One of the main problem in manufacturing processes is the difficulty of measuring critical performance and quality variables in real-time when instrumentation (or sensors) is just not able to do so. When this happens, tests are done to obtain data on how process is performing. In the middle of test results, processes continues without making any adjustments to optimize it, resulting in out-of-spec product events or operational and/or energy inefficiency. In this gap without information one thing is for sure: What is not and cannot be measured, cannot be improved.

Specifically, in the different stages of corn wet milling process are variables for which procedures indicates that central lab test must be done because instrumentation on market does not withstand the process conditions or does not offer required reliability, such as protein or fiber concentrations, sulfur dioxide, **moisture**, etc. Said this, and knowing that there is an important relationship between the variables measured in different stages of the process and those mentioned that cannot or is not recommended to be measured in real time[1], that it seeks to predict these variables with real-time information on the process through Machine Learning (ML) algorithms based on the analysis of large amounts of data to take action and maintain variables under control.

As a result and taking into account Almidones Mexicanos S.A. de C.V. (place of elaboration and implementation of this study, hereinafter ALMEX) needs, as well as the availability of information, infrastructure, previous studies, etc. It was decided that **moisture** prediction at the end of corn Gluten Feed Drying process is where implementation can

[1] Paul Harwood Blanchard. *Technology of Corn Wet Milling and Associated Processes Volume 4 of Industrial chemistry library*. Elsevier, first edition, May 1992. ISBN 0-444-88825-5-3

be carried out more quickly, generating an immediate improvement of operations to this process stage.

The scope of this project is corn gluten feed (co-product in corn derived products) as the product to be studied and the gluten feed drying process from dewatering to Drum dryer output (whose drying medium is hot air) from July 2020 to August 2021.

1.1 Background

ALMEX team has developed an internal study in order to decrease time between gluten feed samples. Current **moisture** analysis is performed every 2 hours by a lab analyst, who takes a grab sample, allows it to cool, and then analyzes it on a benchtop NIR. This sampling frequency has inherent issues with controlling upsets that occur between. This sampling frequency also contributes to poor **moisture** control, as operations must wait to see a change in finish product, after a change to temperature setpoint or another critical variable. Continuous online sampling of the moisture content of finished gluten meal will allow more precise control of finished corn gluten meal **moisture** content.

This studies was comparison between two brands of NIR online instruments with final objective to determine which of their instruments would be the optimal choice for ALMEX needs. Final recommendations in the report[2] were any of both instruments will be good choice to have. However, an important remark in this study is that **regression algorithms were used to calibrate and predict Gluten meal moisture**. Methodology for the study was that samples were analyzed by ALMEX's "in house" method for moisture obtaining, then this results were sent to respective companies to estimate a model that calibrates instruments an predict the output. Predicted values from each instrument were analyzed against the actual values (results on figure 1.1 and figure 1.2).

Referred intern study is a first approach to a concept named *virtual sensing or virtual sensor*, that are techniques to provide economic and feasible alternatives to impractical or costly physical measurements. The virtual sensor is obtained using machine learning techniques by training a predictor whose inputs are the measured variables and the features extracted by a bank of linear observers fed with the same measures. The approach is applicable to infer the value of quantities such as physical states and other time-varying parameters that affect the dynamics of the system.[3] This approach promises cost savings because the company virtually performs physical, costly tasks.

[2] Almidones Mexicanos S.A. de C.V. Comparison study beetween brimrose and perten online nir. Unpublished, dec 2016. Internal report (Almidones Mexicanos S.A. de C.V.)

[3] Daniele Masti, Daniele Bernardini, and Alberto Bemporad. A machine-learning approach to synthesize virtual sensors for parameter-varying systems. *European Journal of Control*, 61:40–49, 2021. ISSN 0947-3580. DOI: https://doi.org/10.1016/j.ejcon.2021.06.005. URL https://www.sciencedirect.com/science/article/pii/S0947358021000637

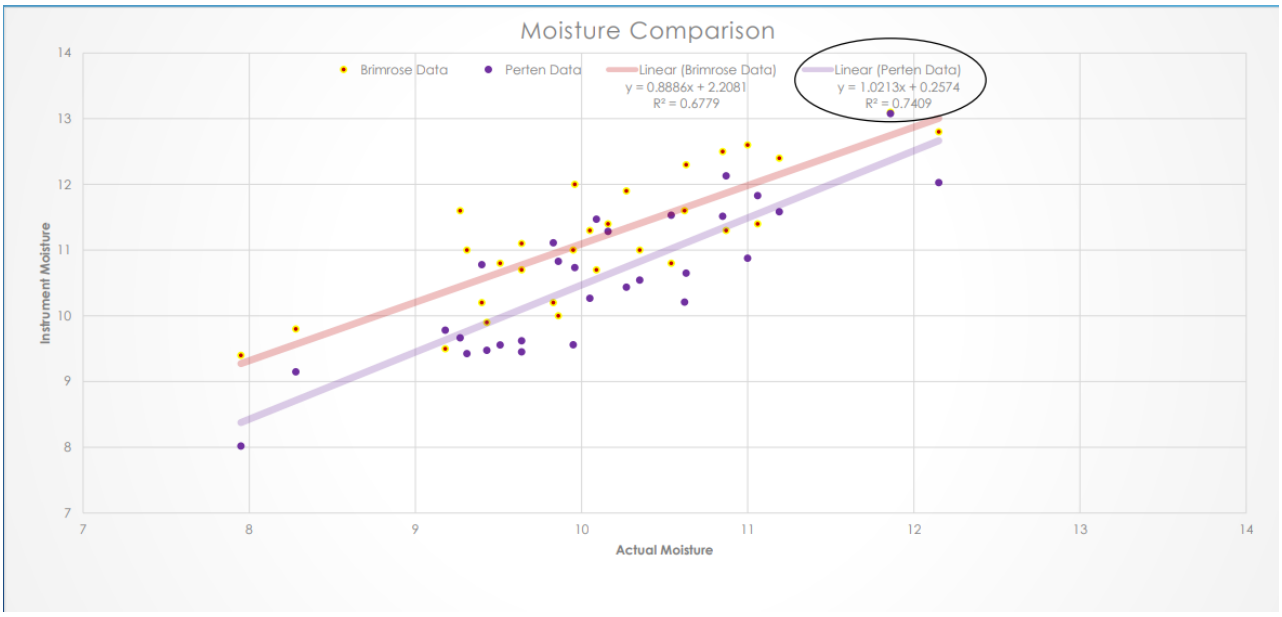

Figure 1.1: Moisture comparison models.

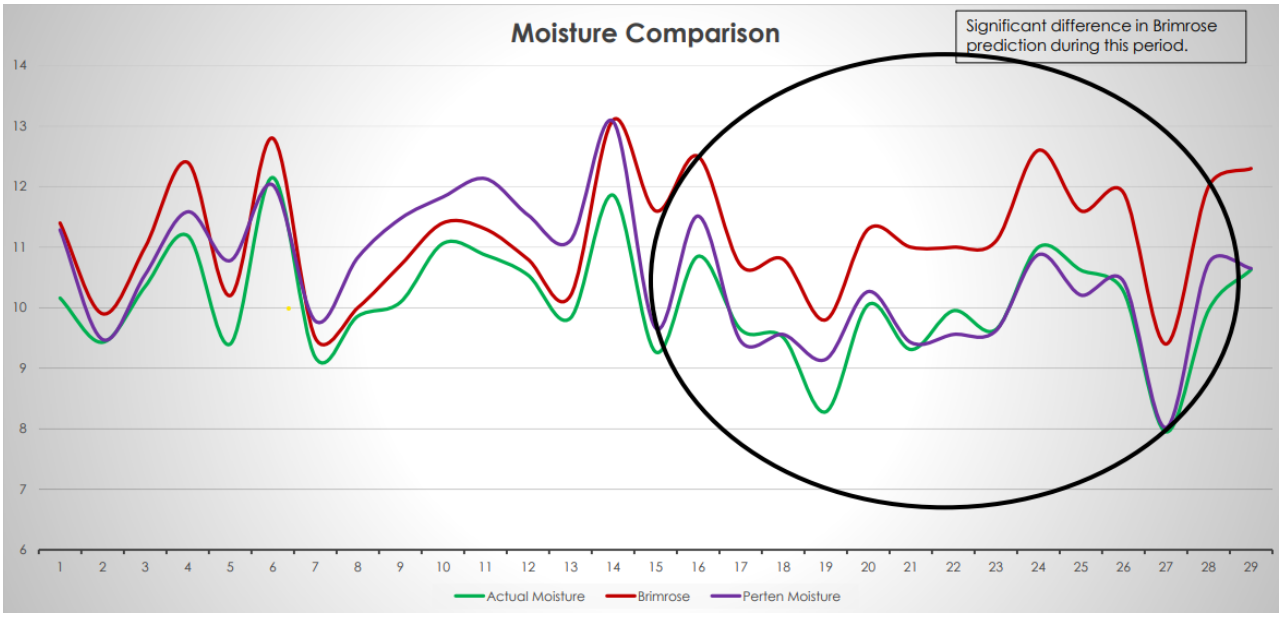

Figure 1.2: Moisture comparison samples.

1.2 Justification

Empirical evidence shows that changing some operational variables on dewatering and drying process affect directly on **moisture**[4], thus, correlation between them must be present. This principle is used in NIR lab and online instrumentation (figure 1.3) that predict moisture using IR absorbance as predictor.

ALMEX internal evaluation about instrumentation for measure **moisture** online show results that does not justify coast to acquire it[5]. In addition, instrumentation does not perform well in hostile environments and these exist throughout dewatering and drying process, therefore, another method to obtain reliable online **moisture** and real-time measure that does not involves physical instrumentation must be evaluated to achieve ALMEX goal. *Virtual sensors* approach is justified because does not requires new hardware or infrastructure investment, is fast to implement and outputs are easy to read with acceptable error determined by the users.

[4] Paul Harwood Blanchard. *Technology of Corn Wet Milling and Associated Processes Volume 4 of Industrial chemistry library*. Elsevier, first edition, May 1992. ISBN 0-444-88825-5-3

Figure 1.3: NIR online instrument example

[5] Almidones Mexicanos S.A. de C.V. Comparison study beetween brimrose and perten online nir. Unpublished, dec 2016. Internal report (Almidones Mexicanos S.A. de C.V.)

1.3 Problem Statement

Currently there is no reliable way in Almidones Mexicanos S.A. de C.V. for the real-time measurement of critical variable **moisture** in gluten feed drying unit operation. Instrumentation found in market for solid products either has a high cost or does not withstand the conditions of the gluten drying process plus gave data each time to time and not in a continuous flow. All this added to the fact that investment in instrumental infrastructure is not contemplated at this time by the organization.

1.4 Aim and Objectives

The aim of this study is to predict **moisture** at gluten feed drying process discharge using measurable variables through instrumentation (sensors) within the process as predictors with an accuracy (understood in ALMEX context as 100 - **MAPE**) greater than 80%.

1.4.1 Specific Objectives

In order to achieve the aim of this work, the following objectives are intended to be developed:

- Get a database with the dependent and independent variables involved in the gluten feed drying process, based on an operational analysis made with operation personnel.

- Find the input random variables that best predict the behavior of the moisture output variable through techniques such as correlation analysis.
- Test different machine learning prediction models such as:
  - Multivariate linear regression models.
  - Support Vector Machine Regression (SVRM).
  - Basic Trees Regression.
  - Bagged Trees Regression.
  - Random Forest Regression.
- Propose a solution to replicate model in operation control screens with real-time data in order operation can take decisions related to moisture prediction.

2 State of the Art

Contents

2.1 Virtual Sensors 23
2.2 Virtual Sensors in industrial applications 24

Today, sensors feed information systems with data describing physical phenomena – such as temperature, pressure, moisture, velocity, chemical components, or material composition – across many areas ranging from industrial applications (e.g., smart factories).

This development of cyber-physical systems is drawing attention to the question of how data can be captured from the physical world and be fed into a connected information systems: the condition of the physical world can either be directly observed (by a physical sensor) or indirectly derived by fusing data from one or more physical sensors, i.e., applying *virtual sensors*.

Typically, embedding physical sensor output into information systems is subject to a number of limitations: equipping assets with sensors is cost-intensive, sensor signals are noisy or may interfere with each other, sensors may lose accuracy over time, or their use is even technically not feasible due to spatial or environmental conditions. They issue signals that aggregate input from physical sensors; thus, they may overcome the limitations mentioned above, offering lower operating cost or increased reliability, agility, or even indirect measurement of physically non-measurable properties. In addition, virtual sensors can make low-level physical sensor information more broadly available for application in cyber-physical systems, but specially, on the level of assets (e.g., replacing or substituting individual sensors)[1].

[1] Dominik Martin, Niklas Kühl, and Gerhard Satzger. Virtual sensors. *Business & Information Systems Engineering*, 63(3): 315–323, 2021. DOI: 10.1007/s12599-021-00689-w

2.1 Virtual Sensors

In general, sensors are technical devices that monitor their environment and continuously produce signals at a regular frequency. A physical sensor is a sensor that reacts to a physical stimulus and transmits

a resulting impulse – typically through electrical signals that can be captured and stored in digital form. In contrast to physical sensors, a so-called *virtual sensor* is a pure software sensor which autonomously produces signals by combining and aggregating signals that it receives (synchronously or asynchronously) from physical or other virtual sensors.

Basic concept of virtual sensors where established by Muir in 1990[2]. He named it because the combination of physical sensors and the data-to-parameter computation produce a measurement of the desired parameter vector, which to the user appears as if were a virtual sensor measuring the parameter vector directly. This parameter vector is a result of a measurement vector, that are the actual output of physical sensor, and a data-to-parameter mapping relation. Figure 2.1 shows this concept more detailed. Nevertheless, there are a number of unresolved challenges (like data access and availability, standardization, platform deployment) limit its application.

[2] P.F. Muir. A virtual sensor approach to robot kinematic identification: theory and experimental implementation. In *1990 IEEE International Conference on Systems Engineering*, pages 440–445, 1990. DOI: 10.1109/ICSYSE.1990.203189

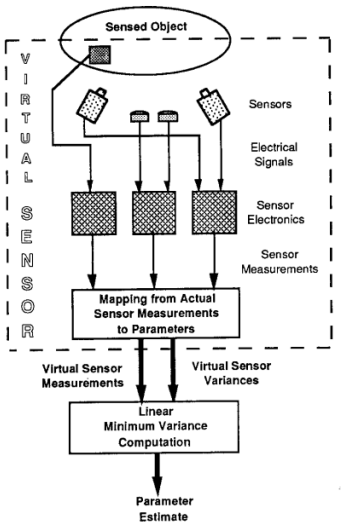

Figure 2.1: The Virtual Sensor Concept.

2.2 *Virtual Sensors in industrial applications*

Industrial applications (logistics, planning, quality control, predictive maintenance, etc.) can benefit from the Virtual Sensor functionalities: increasing the knowledge of the process, reducing the operational costs of the monitoring strategy, and offering a cost-effective solution enhancing monitoring system resilience.

The applications of Virtual Sensor in the manufacturing industry are very heterogeneous. Maschler et al.[3] estimated the combustion duration on a large gas engine using just the rotational speed as input data. They studied in this work the importance of pre-processing the data for greater accuracy, showing different results for the use of Principal Component Analysis. Alonso et al.[4] aimed to calculate the cooling power estimation to enable the replacement of the expensive portable measuring system. They used a model based on a Deep Learning architecture that involved data from the chiller's thermodynamic variables (temperature and pressure) and data from the refrigeration circuit (pressure power).

Other studies focus on the malfunctioning of the system instead. Zenisek et al.[5] presented an approach to stabilise and optimise the metal deposition process, merging information from various sources. The ML-based method generates a valid data stream from heterogeneous sources and can mitigate the problem of data merging through the knowledge of domain experts. Finally, they presented a real use case where they estimated the current weld bead height, one of the principal performance indicators of the process. Ilyas et al.[6] introduced a framework capable of finding sensors in the surrounding environment

[3] Benjamin Maschler, Sören Ganssloser, Andreas Hablizel, and Michael Weyrich. Deep learning based soft sensors for industrial machinery. *Procedia CIRP*, 99: 662–667, 2021. ISSN 2212-8271. DOI: https://doi.org/10.1016/j.procir.2021.03.115. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020

[4] Serafín Alonso, Antonio Morán, Daniel Pérez, Perfecto Reguera, Ignacio Díaz, and Manuel Domínguez. Virtual sensor based on a deep learning approach for estimating efficiency in chillers. In *Communications in Computer and Information Science*, volume 1000, pages 307–319. Springer Verlag, 2019. ISBN 9783030202569. DOI: 10.1007/978-3-030-20257-6_26

[5] Jan Zenisek, Holger Gröning, Norbert Wild, Aziz Huskic, and Michael Affenzeller. Machine learning based data stream merging in additive manufacturing. *Procedia Computer Science*, 200: 1422–1431, 2022. ISSN 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.01.343. 3rd International Conference on Industry 4.0 and Smart Manufacturing

[6] Eushay Bin Ilyas, Marten Fischer, Thorben Iggena, and Ralf Tönjes. Virtual sensor creation to replace faulty sensors using automated machine learning techniques. In *2020 Global Internet of Things Summit (GIoTS)*, pages 1–6, 2020. DOI: 10.1109/GIOTS49054.2020.9119681

and replacing faulty sensors in an automated way. The framework selects the data source based on metadata description, pre-processes historical data, and trains and ranks machine learning algorithms with great results without human intervention. They tested the model predicting the output of a solar power plant.

Tegen et al.[7] proposed Dynamic Intelligent Virtual Sensors (DIVS). The idea was to combine a broader (and not fixed) set of heterogeneous data sources based on Machine Learning to involve the user in the loop. The dynamic part of the concept can be interesting for industrial applications: evaluating the inputs of the Virtual Sensor in terms of information quality (for instance, noise, entropy, etc.) and deciding whether a data source (physical sensor) should be removed or added to the Virtual Sensor.

Virtual Sensor can also be applied in real-time prediction of variables that, for measure it, installing physical sensors is either impractical or the sensor technology is not sufficiently advanced. Dimitrov[8] uses ML-based virtual sensors and showcases three specific examples: blade root bending moment prediction, detection of wind turbine wake center location, and forecasting of blade tip-tower clearance. All examples utilize sequence models (Long Short-Term Memory, LSTM) and use aeroelastic load simulations to generate wind turbine response time series and test model performance. The resulting model performance demonstrates the feasibility of the ML-based virtual sensor approach. Djerioui et al.[9] implemented a Virtual Sensor of the chlorine parameter in water treatment plants using the conductivity, dissolved oxygen, suspended solids, and pH variables as input data. The study compares the performance of a Support Vector Machine (SVM) and an Extreme Learning Machine (ELM) ML algorithm, showing better behaviour using ELM. Pattanayak et al.[10] developed a Virtual Sensors to predict in real-time the Chemical Oxygen Demand (COD) of the river Ganga using the input quality parameters of ammonia, total suspended solids, nitrate, pH, and dissolved oxygen. They evaluated different algorithms, finally building a predictive model based on K-Nearest Neighbours, which was used to predict the water quality at the treatment plant's discharge point. Also, Wastewater treatment is a process where factors such as energy cost or climate footprint are directly related to the process optimisation. Virtual Sensor enables monitoring key parameters in situations where the physical sensors may lead to error due to the constant contact with wastewater. Foschi et al.[11] proposed a Virtual Sensor for the E. Coli value for wastewater disinfection using the data from conventional wastewater physical and chemical indicators (such as COD, nitrate, and ammonia). Their research obtains a predictive model trained using an artificial neural network, which could save up to 57% of disinfectant. Pisa et al.[12] showed a Virtual Sensor to

[7] Agnes Tegen, Paul Davidsson, Radu-Casian Mihailescu, and Jan A. Persson. Collaborative sensing with interactive learning using dynamic intelligent virtual sensors. *Sensors*, 19 (3), 2019. ISSN 1424-8220. DOI: 10.3390/s19030477. URL https://www.mdpi.com/1424-8220/19/3/477

[8] Nikolay Dimitrov and Tuhfe Göçmen. Virtual sensors for wind turbines with machine learning-based time series models. *Wind Energy*, 25(9):1626–1645, 2022. DOI: https://doi.org/10.1002/we.2762. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/we.2762

[9] Mohamed Djerioui, Mohamed Bouamar, Mohamed Ladjal, and Azzedine Zerguine. Chlorine soft sensor based on extreme learning machine for water quality monitoring. *Arabian Journal for Science and Engineering*, 44, 2019. ISSN 2191-4281. DOI: 10.1007/s13369-018-3253-8. URL https://doi.org/10.1007/s13369-018-3253-8

[10] Arunima Sambhuta Pattanayak, Bhawani Shankar Pattnaik, Siba K. Udgata, and Ajit Kumar Panda. Development of chemical oxygen on demand (cod) soft sensor using edge intelligence. *IEEE Sensors Journal*, 20(24):14892–14902, 2020. DOI: 10.1109/JSEN.2020.3010134

[11] Jacopo Foschi, Andrea Turolla, and Manuela Antonelli. Soft sensor predictor of e. coli concentration based on conventional monitoring parameters for wastewater disinfection control. *Water Research*, 191: 116806, 2021. ISSN 0043-1354. DOI: https://doi.org/10.1016/j.watres.2021.116806. URL https://www.sciencedirect.com/science/article/pii/S004313542100004X

[12] Ivan Pisa, Ignacio Santín, Jose Lopez Vicario, Antoni Morell, and Ramon Vilanova. Ann-based soft sensor to predict effluent violations in wastewater treatment plants. *Sensors*, 19 (6), 2019. ISSN 1424-8220. DOI: 10.3390/s19061280. URL https://www.mdpi.com/1424-8220/19/6/1280

predict ammonium and total nitrogen to control effluent violations at the treatment plant using input flow, input ammonium, temperature, and internal recycle flow data. They accomplished the generation of a predictive model using a deep neural network with Long-Short Term Memory neurons, capable of predicting the nitrogen-derived parameters with good accuracy.

Finally, most of the biggest Industrial Automation Companies in the world have their own solution about virtual sensors. Emerson has DeltaV™ Neural (https://www.emerson.com/en-us/catalog/automation-solutions/deltav-ve3153f01) that uses neural network to create virtual sensors for measurements previously available only through the use of lab analysis or online analyzers. Rockwell automation has FactoryTalk Analytics LogixAI (https://www.rockwellautomation.com/en-us/products/software/factorytalk/operationsuite/analytics-logixai.html) that uses predictive analytics to turn raw data about how a system operates into actual decisions about the operation and maintenance of that system. Siemens (https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500/simatic-s7-1500-tm-npu.html) has SIMATIC S7-1500 TM NPU (neural processing unit) module , using neural networks or machine learning algorithms to create virtual sensors.

3 Theoretical Framework

Contents

3.1 Product and Process Description **27**
3.1.1 Corn Gluten Feed 27
3.1.2 Wet Milling Process 28
3.1.3 Drying unit operation 28
3.1.4 Corn Gluten Dewatering and Drying 31
3.2 Machine Learning **31**
3.2.1 Data Preprocessing 32
3.3 Feature Selection Procedures **35**
3.3.1 Stepwise Regression 36
3.3.2 Backward Elimination 38
3.3.3 Principal Component Analysis (PCA) 38
3.3.4 Partial Least Squares Regression (PLS) 40
3.4 Linear Regression **41**
3.4.1 Ordinary Least Squares Method . . . 42
3.5 Support Vector Machines (SVM) **44**
3.6 Decision Trees Regression **46**
3.7 Bagged Tree Algorithm **48**
3.8 Random Forest Tree Algorithm **48**

3.1 Product and Process Description

3.1.1 Corn Gluten Feed

Corn gluten feed is a product derived from corn grain wet milling, an industrial process intended to produce, among others, high fructose syrup for human use. In this process, the soluble part is first separated (Steeping of the corn) and later it is divided by centrifugation into starch and gluten. The latter contains most of the grain endosperm protein (zein), along with small amounts of fiber and starch, not purified in the process. Corn Gluten is available in either wet or dry forms and it is main used as cattle feed[1]. In figure 3.1 appear corn gluten presentation and uses.

[1] Paul Harwood Blanchard. *Technology of Corn Wet Milling and Associated Processes Volume 4 of Industrial chemistry library*. Elsevier, first edition, May 1992. ISBN 0-44488825-5-3

Figure 3.1: Corn gluten feed applications

3.1.2 *Wet Milling Process*

Corn wet milling process involves chemical, biochemical and mechanical operations to separate the grain into its main components: starch, gluten, germ and fiber (figure 3.4). The process begins with grain steeping in order to soften it, followed by grinding and separation operations. The fractions have different physical properties, so they can be separated by methods based on differences in density and particle size. The basic characteristic of this industry is to achieve the separation of the main components of corn using large amounts of water.

First, corn is cleaned to remove foreign material. The grain is transported to tanks where it is soaked in an aqueous Sulfur Dioxide (SO) solution for a period of 2 to 3 days at a temperature of approximately 50°C. During the steeping process, about 6% of the grain's dry weight dissolves. After this process, the corn grain absorbed abundant water and reacted with SO, making it soft enough to be disintegrated even by simple friction with the fingers.

Subsequently, a first rough grinding is carried out on toothed rotating discs mills. The distance between teeth is variable depending on the size of the grains. The whole germ comes off and is separated by flotation, dried and destined for the production of oil by means of pressing and extraction with hexane; the crude oil is subsequently refined. After separation of the germ, the resulting suspension is ground in an impact mill to pulverize the endosperm particles, while leaving the fibrous material intact. The suspension is filtered through a series of sieves with a decreasing mesh size (75 to 50 μm), impacting the suspension on the last sieve to allow the passage of starch and gluten.

Starch and gluten are separated by centrifugation. The less dense suspension corresponds to the gluten, which is concentrated and subsequently dehydrated by means of rotary filters, from which a gluten cake is obtained that is dried with a hot air flow. On the other hand, starch suspension or "slurry" emerging from the lower end of the hydrocyclone is concentrated and backwashed. This starch suspension is used in different final products processes. These may be modified starches, hydrolysis processes for the production of syrups or simply their dehydration in a fluidized bed with hot air[2]. This process is summarized through a diagram block in figure 3.3.

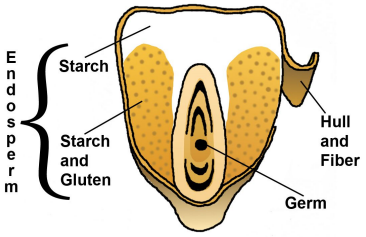

Figure 3.2: Corn composition

[2] Paul Harwood Blanchard. *Technology of Corn Wet Milling and Associated Processes Volume 4 of Industrial chemistry library*. Elsevier, first edition, May 1992. ISBN 0-44488825-5-3

3.1.3 *Drying unit operation*

Drying is described as a process of removing volatile substances (**moisture**) to produce a dry, solid product[3]. **Moisture** appears as a liquid solution within the solid, that is, in its micro-structure. When a wet solid is subjected to thermal drying, two processes will occur simultaneously:

[3] Christie J. Geankoplis, A. Allen Hersel, and Daniel H. Lepek. *Transport processes & separation process principles*. Pearson, fifth edition, Jan 2018. ISBN 978-0-13-418102-8

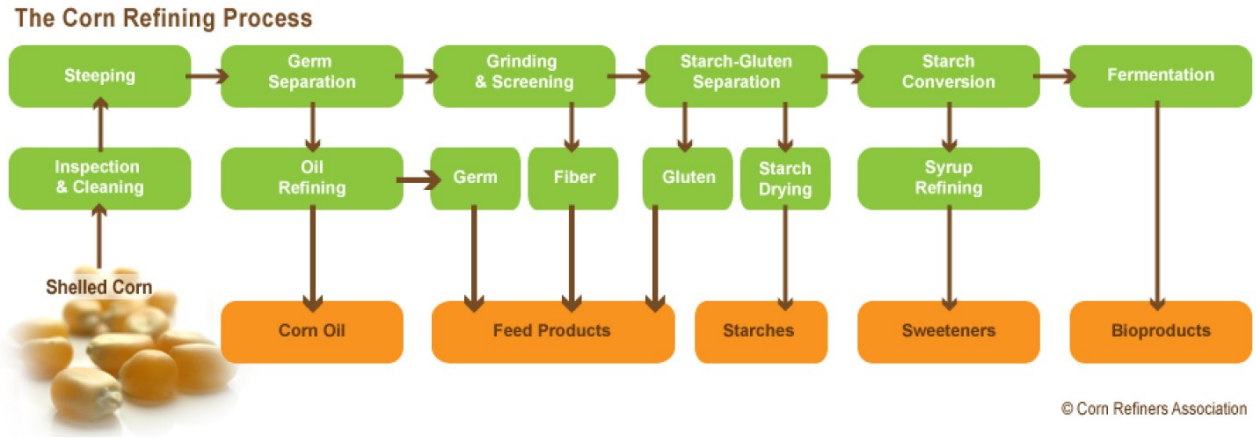

Figure 3.3: Wet milling process block diagram

1. There will be energy transfer (commonly as heat) from the surroundings to evaporate moisture from the surface.

2. There will be transfer of internal moisture to the surface of the solid.

The rate at which drying is performed is determined by the rate at which these two processes are carried out. The energy transfer, in the form of heat, from the surroundings to the wet solid can occur as a result of convection, conduction, and/or radiation, or a combination of these. This process is carried out in equipment called dryers, which can be:

Direct heating dryers:

- Tray dryers.
- Fluidized bed dryers
- Tunnel dryers.
- Vibrating bed dryers.
- Spray dryers.
- Rotary dryers.

Indirect heated dryers:

- Vacuum tray dryers.
- Atmospheric pressure tray dryers.
- Freeze dryers.

- Drum dryers.
- Dryers with circulation through the bed.

This study is developed with variables obtained from a direct heating rotary dryer, which consists of a drum that rotates on its central axis through which the material to be dried circulates (a typical schema of a rotary dryer can be seen in figure **??**. A stream of hot air is introduced through the interior of the drum, which will simultaneously be the means of transmitting heat and a vehicle for transporting humidity.

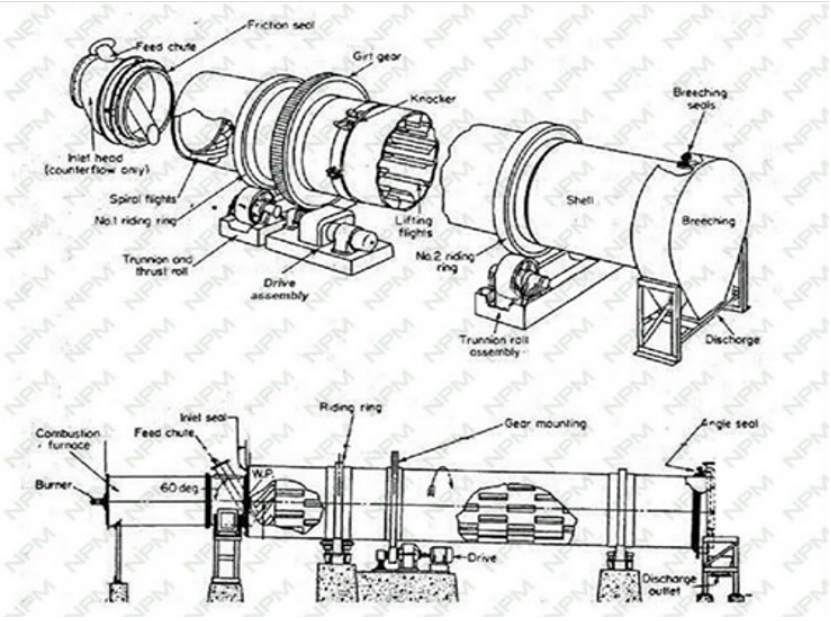

Figure 3.4: Drum Dryer typical schema

Air or combustion gases can be circulated inside the drum either in parallel or counter-current to the flow of the material to be dried. The solid is transported from one end of the drum to the other through a small unevenness of the cylinder that displaces the product by sliding on the inner surface of the drum. The dryers are also equipped with internal flights that lift the material and drop it by gravity as the drum rotates. Part of the material, fines, are dragged by the air current from which they are removed by means of a cyclone separator located at the air outlet. The variables that affect drying in a dryer of this type are: temperature, humidity and air velocity; permanence of the material inside the dryer, which will depend on the speed of rotation of the cylinder and its inclination; number of fins that the drum has and the particular characteristics of the material to be dried, size, porosity, density, etc.

3.1.4 Corn Gluten Dewatering and Drying

Prior to drying, gluten must be dewatered mechanically. This happens on an rotary drum filter (figure 3.5). This equipment uses a segmented rotary drum, which is covered with a belt as the filtering surface. The Drum dips into a trough of thickened gluten, and vacuum is applied to build up a cake. As each segment rotates clear of the trough the cake is sucked free of surplus moisture and is discharged where the belt is pulled away from the drum by a discharge roller. This filter is able to produce a gluten cake with 40-43% dry substance, which make final drying more efficient.

Gluten drying has typically done with a flash dryer with direct gas firing, steam heated tubular dryers and rotary dryers. Final decision on which equipment is driven by fuel and operating cost[4].

Almex uses a rotary dryer and six rotary vacuum filters. Gluten cake from filters enters to a set of hammer mills (crushers) to avoid clumps, but these crushers can be bypassed in case of equipment malfunction. The wet feed from crushers enters the south side of the dryer and falls into the drum. The discharge exits on the north side to a screw conveyor, which discharges to a drag conveyor that feeds the gluten mills (to standardize product granularity). The output of these mills enters a pneumatic conveyor to finally fall into the gluten storage hopper.

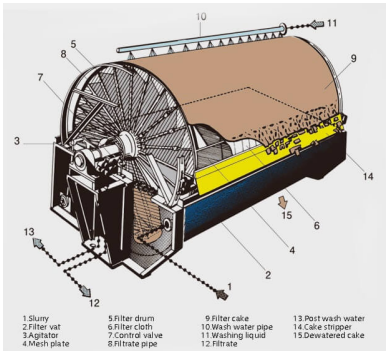

Figure 3.5: Rotary Drum Filter

[4] Paul Harwood Blanchard. *Technology of Corn Wet Milling and Associated Processes Volume 4 of Industrial chemistry library*. Elsevier, first edition, May 1992. ISBN 0-4448825-5-3

3.2 Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.[5] The basic concept in data science involves using statistical learning and optimization methods that let computers analyze datasets and identify patterns leveraging data mining to identify historic trends and inform future models.

[5] IBM Cloud Education. Machine learning. https://www.ibm.com/cloud/learn/machine-learning, July 2020

Within artificial intelligence (AI) and machine learning, there are two basic approaches:

- **Supervised learning problems:** Applications in which the data comprises examples of the inputs along with their corresponding targets are known. Cases in which the aim is to assign each input to one of a finite number of discrete categories, are called *classification problems* (such as Logistic regression, Decision trees classifier, Support Vector Machine, etc). On the other hand, if the desired output consists of one or more continuous variables, then the task is called *regression* (Linear Regression, Decision Tree Regression, Random Forest Regression, Support Vector Machine Regression, etc).
- **Unsupervised learning problems:** Applications in which the data

consists of a set of input without any corresponding target values. The goal may be to discover groups of similar examples within the data, or to determine the distribution of data within the input space, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization. These algorithms discover hidden patterns or data groupings without the need for human intervention.

Also, there is a third approach, the technique of **reinforcement learning**[6], which is concerned with the problem of finding suitable actions to take in a given situation in order to maximize a reward. Here the learning algorithm is not given examples of optimal outputs, in contrast to supervised learning, but must instead discover them by a process of trial and error. Typically there is a sequence of states and actions in which the learning algorithm is interacting with its environment. In many cases, the current action not only affects the immediate reward but also has an impact on the reward at all subsequent time steps.

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, NY, first edition, May 2006. ISBN 0-387-31073-8

Finally, UC Berkeley School of Information[7] breaks out, in order to explain in an easier way how machine learning works, the learning system of a machine learning algorithm into three main parts:

[7] UC Berkeley School of Information. What is machine learning (ml)? https://ischoolonline.berkeley.edu/blog/what-is-machine-learning, June 2020

1. **Decision Process**: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labeled or unlabeled, your algorithm will produce an estimate about a pattern in the data.

2. **Loss or Cost function**: Function that evaluates the prediction of the model. If there are known examples, a cost function can make a comparison to assess the accuracy of the model.

3. **Model Optimization Process**: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this "evaluate and optimize" process, updating weights autonomously until a threshold of accuracy has been met.

3.2.1 *Data Preprocessing*

Data Preprocessing is the step in which data gets transformed, or Encoded, to bring it to such a state that now the machine learning model can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine

learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Different models have different sensitivities to the type of predictors in the model; how the predictors enter the model is also important. Transformations of the data to reduce the impact of data skewness or outliers can lead to significant improvements in performance.

Preprocessing of data is mainly to check the data quality (Data profiling). The quality can be checked by the following:

- **Accuracy:** To check whether the data entered is correct or not.
- **Completeness:** To check whether the data is available or not recorded.
- **Consistency:** To check whether the same data is kept in all the places that do or do not match.
- **Timeliness:** The data should be updated correctly.
- **Believability:** The data should be trustable.
- **Interpretability:** The understandability of the data.

To complete a correct methodology for data preprocessing, it is recommended to follow next major tasks[8]:

1. **Data cleansing:** process to remove incorrect data, incomplete data and inaccurate data from the datasets, and it also replaces the missing values. There are some techniques in data cleaning:

   Handling missing value:

   - Standard values like "Not Available" or "NA" can be used to replace the missing values.
   - Missing values can also be filled manually but it is not recommended when that dataset is big.
   - Attribute's mean value can be used to replace the missing value when the data is normally distributed wherein in the case of non-normal distribution median value of the attribute can be used.
   - While using regression or decision tree algorithms the missing value can be replaced by the most probable value.

   Handling noise (random error or containing unnecessary data points):

   - Binning.

[8] Suad A. Alasadi and Wesam S. Bhaya. Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12:4102–4107, sep 2017. DOI: 10.3923/jeasci.2017.4102.4107

- Regression.
- Clustering.

2. **Data integration:** Process of combining multiple sources into a single dataset. There are some problems to be considered during data integration:

   - **Schema integration:**Integrates metadata(a set of data that describes other data) from different sources.
   - **Entity identification problem:** Identifying entities from multiple databases.
   - **Detecting and resolving data value concepts:** The data taken from different databases while merging may differ.

3. **Data reduction:** Process that helps in the reduction of the volume of the data which makes the analysis easier yet produces the same or almost the same result. This reduction also helps to reduce storage space:

   - **Dimensionality reduction:** In this process, the reduction of random variables or attributes is done so that the dimensionality of the data set can be reduced. Combining and merging the attributes of the data without losing its original characteristics.
   - **Numerosity Reduction:** The representation of the data is made smaller by reducing the volume. There will not be any loss of data in this reduction.
   - **Data compression:** The compressed form of data is called data compression. This compression can be lossless or lossy. When there is no loss of information during compression it is called lossless compression. Whereas lossy compression reduces information but it removes only the unnecessary information.

4. **Data Transformation:** Change made in the format or the structure of the data. There are some methods in data transformation:

   - **Smoothing:** Remove noise and helps in knowing the important features of the dataset.
   - **Aggregation:** Data is stored and presented in the form of a summary. The dataset which is from multiple sources is integrated into with data analysis description.
   - **Discretization:** The continuous data here is split into intervals. Discretization reduces the data size.
   - **Normalization:** It is the method of scaling the data so that it can be represented in a smaller range.

3.3 Feature Selection Procedures

Determining which predictors should be included in a model is becoming one of the most critical questions as data are becoming increasingly high-dimensional. A model with less predictors may be more interpretable and less costly especially if there is a cost to measuring the predictors. Statistically, it is more attractive to estimate fewer parameters. Also, some models may be negatively affected by non-informative predictors.

Many models, especially those based on regression slopes and intercepts, will estimate parameters for every term in the model. Because of this, the presence of non-informative variables can add uncertainty to the predictions and reduce the overall effectiveness of the model.Feature selection is primarily focused on removing non-informative or redundant predictors from the model.

One way to think about feature selection methods are in terms of supervised and unsupervised methods. The difference has to do with whether features are selected based on the target variable or not. Unsupervised feature selection techniques ignores the target variable, such as methods that remove redundant variables using correlation. Supervised feature selection techniques use the target variable, such as methods that remove irrelevant variables.

Another way to consider the mechanism used to select features which may be divided into wrapper and filter methods. These methods are almost always supervised and are evaluated based on the performance of a resulting model on a hold out dataset. Wrapper feature selection methods create many models with different subsets of input features and select those features that result in the best performing model according to a performance metric.Filter feature selection methods use statistical techniques to evaluate the relationship between each input variable and the target variable, and these scores are used as the basis to choose (filter) those input variables that will be used in the model.

Finally, there are some machine learning algorithms that perform feature selection automatically as part of learning the model. These techniques can be referred as intrinsic feature selection methods. This includes algorithms such as penalized regression models like Lasso and decision trees, including ensembles of decision trees like random forest[9].

Feature selection is also related to dimensionally reduction techniques in that both methods seek fewer input variables to a predictive model. The difference is that feature selection select features to keep or remove from the dataset, whereas dimensionality reduction create a projection of the data resulting in entirely new input features (such as PCA).

[9] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

Feature selection procedures can be summarized as follows:

- **Unsupervised**: Do not use the target variable (e.g. remove redundant variables).
  - Correlation.
- **Supervised**: Use the target variable (e.g. remove irrelevant variables).
  - *Wrapper*: Search for well-performing subsets of features.
    - Recursive Feature Elimination.
    - Forward Selection.
    - Backward Elimination.
    - Stepwise Regression.
  - *Filter*: Select subsets of features based on their relationship with the target.
    - Statistical Methods.
      - Pearson's Correlation.
    - Feature Importance Methods.
      - Gini Importance.
      - Permutation Feature Importance.
    - *Intrinsic*: Algorithms that perform automatic feature selection during training.
      - Decision Trees.

Chosen feature selection procedures for this study where based on models used to predict corn gluten moisture and Dataset analysis (Correlation between predictors, Wrapper procedures such as Backward elimination and stepwise regression, Dimensionality Reduction procedures such as principal component analysis and partial least squares regression). Theory will be explained on this section. Feature Selection criteria will be discuss on section 3.4.

3.3.1 *Stepwise Regression*

The stepwise regression procedure starts off by choosing an equation containing the single best **x** variable and then attempts to build up with subsequent additions of **x**'s one at a time as long as these additions are worthwhile. The order of addition is determined by using the partial t-test values (or equivalent F-test) to select which variable should enter next. The highest partial t-value is compared to a (selected or default) t-to-enter value. After a variable has been added, the equation is examined to see if any variable should be deleted. Usually done on a computer software [10], the basic procedure is as follows:

[10] William Mendenhall and Terry Sincich. *A second course in statistics : regression analysis*. Prentice Hall, seventh edition, 2012. ISBN 978-0-321-69169-9

1. The software program fits all possible one-variable models of the form:

$$E(y) = w_0 + w_1 x_i$$

to the data, where x_i is the ith independent variable, $i = 1, ..., \mathcal{D}$. For each model, the test of the null hypothesis:

$$H_0 : w_1 = 0$$

against the alternative hypothesis:

$$H_a : w_1 \neq 0$$

is conducted using the t-test for a single w parameter. The independent variable that produces the largest (absolute) t-value is declared the best one-variable predictor of y. Call this independent variable x_1.

2. The stepwise program now begins to search through the remaining $(\mathcal{D} - 1)$ independent variables for the best two-variable model of the form:

$$E(y) = w_0 + w_1 x_1 + w_2 x_i$$

This is done by fitting all two-variable models containing x_1 (the variable selected in the first step) and each of the other $(\mathcal{D} - 1)$ options for the second variable x_i. The t-values for the test $H_0 : w_2 = 0$ are computed for each of the $\mathcal{D} - 1$ models (corresponding to the remaining independent variables, $x_i, i = 2, 3, ..., \mathcal{D}$), and the variable having the largest t is retained. Call this variable x_2. Before proceeding to Step 3, the stepwise routine will go back and check the t-value of $\hat{w}_1$ after $\hat{w}_2 x_2$ has been added to the model. If the t-value has become non-significant at some specified α level (say α = .05), the variable x_1 is removed and a search is made for the independent variable with a w parameter that will yield the most significant t-value in the presence $\hat{w}_2 x_2$. The reason the t-value for x_1 may change from step 1 to step 2 is that the meaning of the coefficient $\hat{w}_1$ changes. In step 2, we are approximating a complex response surface in two variables with a plane. The best-fitting plane may yield a different value for $\hat{w}_1$ than that obtained in step 1. Thus, both the value of $\hat{w}_1$ and its significance usually changes from step 1 to step 2. For this reason, stepwise procedures that recheck the t-values at each step are preferred.

3. The stepwise regression procedure now checks for a third independent variable to include in the model with x_1 and x_2. That is, we seek the best model of the form:

$$E(y) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_i$$

To do this, the computer fits all the $(\mathcal{D} - 2)$ models using x_1, x_2, and each of the $(\mathcal{D} - 2)$ remaining variables, x_i, as a possible x_3. The criterion is again to include the independent variable with the largest t-value. Call this best third variable x_3. The better programs now recheck the t-values corresponding to the x_1 and x_2 coefficients, replacing the variables that yield non-significant t-values. This procedure is continued until no further independent variables can be found that yield significant t-values (at the specified α level) in the presence of the variables already in the model.

The result of the stepwise procedure is a model containing only those terms with t-values that are significant at the specified α level.

3.3.2 *Backward Elimination*

The backward elimination method is also an economical procedure. It tries to examine only the "best" regressions containing a certain number of variables.

The basic steps in the procedure are these[11]:

[11] Norman R. Draper and Harry Smith. *Applied Regression Analysis*. Wiley-Interscience, third edition, 1998. ISBN 0-471-17082-8

1. A regression equation containing all variables is computed:

$$E(y) = w_0 + w_1 x_1 + + w_2 x_2 + \cdots + + w_{\mathcal{D}} x_{\mathcal{D}}$$

2. The partial t-test (or F) value is calculated for every predictor variable (testing $H_0 : w = 0$ versus $H_a : w \neq 0$) treated as though it were the last variable to enter the regression equation.

3. The lowest partial t-test value, say, t_L., is compared with a preselected or default significance level, say, α = .05:

   (a) If $t_L < \alpha$, remove the variable x_L, which gave rise to t_L, from consideration and recompute the regression equation in the remaining variables; reenter stage (2).

   (b) If $t_L > \alpha$, adopt the regression equation as calculated.

3.3.3 *Principal Component Analysis (PCA)*

PCA (Principal Component Analysis) is a multivariate technique that analyzes a data table in which observations are described by several inter-correlated quantitative dependent variables[12]. Is a commonly

[12] Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010. ISSN 1939-0068. DOI: 10.1002/wics.101. URL http://dx.doi.org/10.1002/wics.101

used data reduction technique. This method seeks to find linear combinations of the predictors, known as principal components (PCs), which capture the most possible variance. The first PC is defined as the linear combination of the predictors that captures the most variability of all possible linear combinations. Then, subsequent PCs are derived such that these linear combinations capture the most remaining variability while also being uncorrelated with all previous PCs. Mathematically, the jth PC can be written as:

$$PC_j = (a_{j1} \times \text{Predictor 1}) + (a_{j2} \times \text{Predictor 2}) + \cdots + (a_{j\mathcal{D}} \times \text{Predictor}\mathcal{D})$$

$\mathcal{D}$ is the number of predictors. The coefficients $a_{j1}, a_{j2}, \ldots, a_{j\mathcal{D}}$ are called component weights and help us understand which predictors are most important to each PC.

The primary advantage of PCA, is that it creates components that are uncorrelated. As we know, some predictive models prefer predictors to be uncorrelated, or at least low correlation, in order to find solutions and to improve the model's numerical stability. PCA preprocessing creates new predictors with desirable characteristics for these kinds of models.

While PCA delivers new predictors with desirable characteristics, it must be used with understanding and care. Caveats to keep in mind prior using PCA to dimension reduction [13] are:

1. PCA seeks predictor-set variation without regard to any further understanding of the predictors (i.e., measurement scales or distributions) or to knowledge of the modeling objectives (i.e., response variable). Hence, PCA can generate components that summarize characteristics of the data that are irrelevant to the underlying structure of the data and also to the ultimate modeling objective and make it less interpretable.

2. PCA seeks linear combinations of predictors that maximize variability, it will naturally first be drawn to summarizing predictors that have more variation. If the original predictors are on measurement scales that differ in orders of magnitude , then the first few components will focus on summarizing the higher magnitude predictors, while latter components will summarize lower variance predictors. This means that the PC weights will be larger for the higher variability predictors on the first few components.To help PCA avoid summarizing distributional differences and predictor scale information, it is best to first transform skewed predictors and then center and scale the predictors prior to performing PCA. Centering and scaling enables PCA to find the underlying relationships in the data without being influenced by the original measurement scales.

[13] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

3. PCA is that it does not consider the modeling objective or response variable when summarizing variability. Because PCA is blind to the response, it is an unsupervised technique. If the predictive relationship between the predictors and response is not connected to the predictors variability, then the derived PCs will not provide a suitable relationship with the response.

Pre-processing predictors via PCA prior to performing regression is known as principal component regression (PCR). PCR solves the collinearity problem and the ability to eliminate the lesser principal components allows some noise (random error) reduction. However, PCR is a two step method and thereby has the risk that useful (predictive) information will end up in discarded principal components and that some noise will remain in the components used for regression[14].

[14] Paul Geladi and Bruce R. Kowalski. Partial least-squares regression: a tutorial. *Anal. Chim. Acta*, 185(C):1–17, 1986. DOI: 10.1016/0003-2670(86)80028-9

3.3.4 *Partial Least Squares Regression (PLS)*

The PLS model is built on the properties of the NIPALS algorithm. Subsequently, NIPALS method was adapted for the regression setting with correlated predictors and called this adaptation "PLS"[15]. Briefly, the NIPALS algorithm iteratively seeks to find underlying, or latent, relationships among the predictors which are highly correlated with the response. For a univariate response, each iteration of the algorithm assesses the relationship between the predictors $\mathbf{x}$ and response y and numerically summarizes this relationship with a vector of weights $\mathbf{w}$; this vector is also known as a *direction*. The predictor data are then orthogonally projected onto the direction to generate scores ($\mathbf{t}$). The scores are then used to generate loadings ($\mathbf{p}$), which measure the correlation of the score vector to the original predictors. At the end of each iteration, the predictors and the response are "deflated" by subtracting the current estimate of the predictor and response structure, respectively. The new deflated predictor and response information are then used to generate the next set of weights, scores, and loadings. These quantities are sequentially stored in matrices $\mathbf{W}$, $\mathbf{T}$, and $\mathbf{P}$, respectively, and are used for predicting new samples and computing predictor importance.A schematic of the PLS relationship between predictors and the response can be seen in figure 3.6.

[15] S. Wold, H. Martens, and H. Wold. The multivariate calibration problem in chemistry solved by the pls method. In Bo Kågström and Axel Ruhe, editors, *Matrix Pencils*, pages 286–293, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg. ISBN 978-3-540-39447-1

To obtain a better understanding of the algorithm's function, it can be linked it to well-known statistical concepts of covariance and regression. In particular, Stone and Brooks[16] showed that like PCA, PLS finds linear combinations of the predictors. These linear combinations are commonly called components or latent variables. While the PCA linear combinations are chosen to maximally summarize predictor space variability, the PLS linear combinations of predictors are chosen

[16] M. Stone and R. J. Brooks. Continuum regression: Cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(2):237–258, 1990. DOI: https://doi.org/10.1111/j.2517-6161.1990.tb01786.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1990.tb01786.x

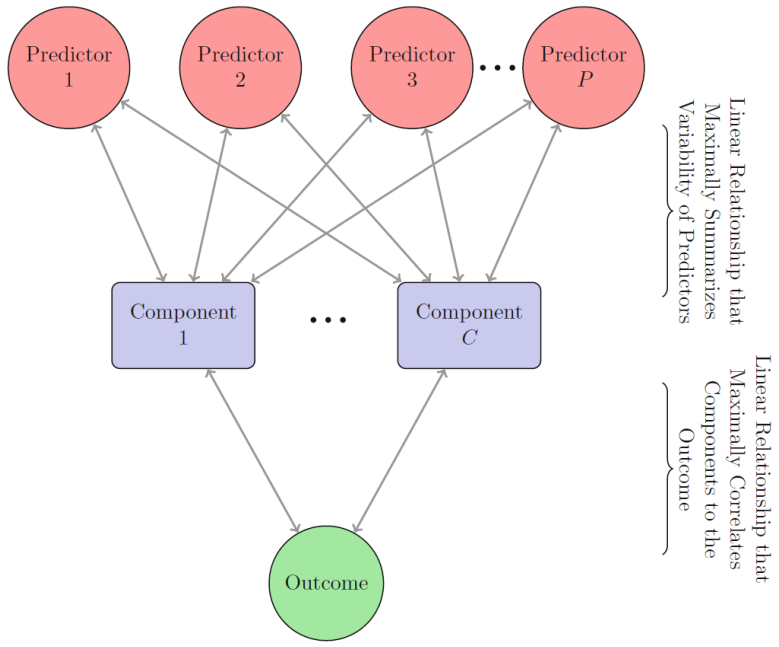

Figure 3.6: A diagram depicting the structure of a PLS model. PLS finds components that simultaneously summarize variation of the predictors while being optimally correlated with the outcome

to maximally summarize covariance with the response. This means that PLS finds components that maximally summarize the variation of the predictors while simultaneously requiring these components to have maximum correlation with the response. PLS therefore strikes a compromise between the objectives of predictor space dimension reduction and a predictive relationship with the response. In other words, PLS can be viewed as a supervised dimension reduction procedure; PCR is an unsupervised procedure.

Prior to performing PLS, the predictors should be centered and scaled, especially if the predictors are on scales of differing magnitude. As described above, PLS will seek directions of maximum variation while simultaneously considering correlation with the response. Even with the constraint of correlation with the response, it will be more naturally drawn towards predictors with large variation. Therefore, predictors should be adequately pre-processed prior to performing PLS[17].

[17] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

3.4 *Linear Regression*

The goal of regression models is to predict the value of one or more continuous target variables t given the value of a $\mathcal{D}$-dimensional vector $\mathbf{x}$ of *input* variables. **Linear regression model** are which have the property of being linear functions of their adjustable parameters.

Given a training data set comprising N observations $\{\mathbf{x}_n\}$, where $n = 1, ..., N$, together with corresponding target values $\{t_n\}$, the goal is to predict the value of t for a new value of $\mathbf{x}$. In the simplest approach, this can be done by directly constructing an appropriate function $y(\mathbf{x})$ whose values for new inputs $\mathbf{x}$ constitute the predictions for the corresponding values of t.

Although linear models have significant limitations as practical techniques for pattern recognition, particularly for problems involving input spaces of high dimensionality, they have nice analytical properties and form the foundation for more sophisticated models[18].

[18] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, NY, first edition, May 2006. ISBN 0-387-31073-8

The simplest linear model for regression is one that involves a linear combination of the input variables:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D \tag{3.1}$$

where $\mathbf{x} = (x_1, ..., x_D)^T$. This is often simply known as **linear regression**. The key property of this model is that it is a linear function of the parameters $w_0, ..., w_D$. It is also, however, a linear function of the input variables x_i, and this imposes significant limitations on the model. We therefore extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables, of the form:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \tag{3.2}$$

where $\phi_j(\mathbf{x})$ are known as *basis functions*. By denoting the maximum value of the index j by $M - 1$, the total number of parameters in this model will be M . The parameter w_0 allows for any fixed offset in the data and is sometimes called a bias parameter. It is often convenient to define an additional dummy 'basis function' $\phi(\mathbf{x}) = 1$ so that:

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \tag{3.3}$$

where $\mathbf{w} = (w_0, ..., w_{M-1})^T$ and $\phi = (\phi_0, ..., \phi_{M-1})^T$. If the original variables comprise the vector $\mathbf{x}$, then the features can be expressed in terms of the basis functions $\{\phi_j(\mathbf{x})\}$. For this work, basis function is simply the identity $\phi(\mathbf{x}) = \mathbf{x}$.

3.4.1 *Ordinary Least Squares Method*

Functions of the form (3.2) are called linear models, however, because this function is linear in w. It is this linearity in the parameters that will greatly simplify the analysis of this class of models.

The values of the coefficients will be determined by fitting the linear model to training data. This can be done by **minimizing an error function** that measures the misfit between the function $y(\mathbf{x}, \mathbf{w})$, for any given value of $\mathbf{w}$, and the training set data points. We can represent said affirmation as:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \tag{3.4}$$

In therms of (3.3):

$$t = \mathbf{w}^T \mathbf{x} + \epsilon \tag{3.5}$$

where ϵ is a zero mean Gaussian random variable.

One simple choice of error function, which is widely used, is given by the **sum of the squares of the errors (SSE)** between the predictions $y(\mathbf{x}, \mathbf{w})$ for each data point x_n and the corresponding target values t_n:

$$SSE = E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^T \mathbf{x}_n\}^2 \tag{3.6}$$

Minimizing it:

$$E(W) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^T \mathbf{x}_n\}^2 = (t - \mathbf{w}^T X)^T (t - \mathbf{w}^T X) \rightarrow \min_{\mathbf{w}^T} \tag{3.7}$$

The resulting estimator of $\mathbf{w}$ is:

$$\hat{\mathbf{w}^T} = (X^T X)^{-1} X^T t \tag{3.8}$$

The term **Ordinary Least Squares (OLS)** comes from the fact that these estimates minimize the sum of squared of the errors[19].

The required conditions of the OLS estimator are[20]:

1. *Linearity:*

   $t_n = \mathbf{w}^T \mathbf{x}_n + \epsilon_n$ and $E[\epsilon_n] = 0$

   Assumes that the functional relationship between dependent and explanatory variables is linear in parameters, that the error term enters additively and that the parameters are constant across individuals n.

2. *Independence:*

   $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$ i.i.d. (independent and identically distributed)

   This assumption is in practice guaranteed by random sampling.

3. *Exogeneity:*

   (a) $\epsilon_n | \mathbf{x}_n \sim \mathcal{N}(0, \sigma_n^2)$

[19] Andrius Buteikis. Practical econometrics and data science. http://web.vu.lt/mif/a.buteikis/wp-content/uploads/PE_Book/3-2-OLS.html, October 2020

[20] Kurt Schmidheiny. The multiple linear regression model. https://www.schmidheiny.name/teaching/ols.pdf, October 2022

(b) $\epsilon_n \perp\!\!\!\perp \mathbf{x}_n$ (independent)

(c) $E[\epsilon_n|\mathbf{x}_n] = 0$ (mean independent)

(d) $Cov[\mathbf{x}_n, \epsilon_n] = 0$ (uncorrelated)

(a) assumes that the error term is normally distributed conditional on the explanatory variables. (b) means that the error term is independent of the explanatory variables. (c) states that the mean of the error term is independent of the explanatory variables. (d) means that the error term and the explanatory variables are uncorrelated.

4. *Error Variance:*

(a) $V[\epsilon_n|\mathbf{x}_n] = \sigma^2 < \infty$ (homoscedasticity)

(b) $V[\epsilon_n|\mathbf{x}_n] = \sigma^2 = g(\mathbf{x}_n) < \infty$ (conditional heteroscedasticity)

Homoscedasticity means that the variance of the error term is a constant. Conditional Heteroscedasticity allows the variance of the error term to depend on the explanatory variables.

5. *Identifiability:*

$E[\mathbf{x}_n^T \mathbf{x}_n] = Q_{xx}$ is positive and finite
$rank(X) = D + 1 < N$

Assumes that the regressors are not perfectly collinear, i.e. no variable is a linear combination of the others. For example, there can only be one constant. Intuitively, means that every explanatory variable adds additional information. Also assumes that all regressors (but the constant) have strictly positive variance both in expectations and in the sample and not too many extreme values.

3.5 Support Vector Machines (SVM)

SVMs are a class of powerful, highly flexible modeling techniques. Is an extension of the support vector machine classifier that results from enlarging the feature space in a specific way, using kernels. The theory behind SVMs was originally developed in the context of classification models [21].

An important property of support vector machines is that the determination of the model parameters corresponds to a convex optimization problem, and so any local solution is also a global optimum[22].

Recall that linear regression seeks to find parameter estimates that minimize SSE (regularized) given by:

[21] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

[22] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, NY, first edition, May 2006. ISBN 0-387-31073-8

$$\frac{1}{2}\sum_{n=1}^{N}\{y_n - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{3.9}$$

To obtain sparse solutions, the quadratic error function is replaced by an *ϵ-insensitive error function*[23], which gives zero error if the absolute difference between the prediction $y(\mathbf{x})$ and the target t is less than ϵ where $\epsilon > 0$. A simple example of an ϵ-insensitive error function, having a linear cost associated with errors outside the insensitive region, is given by:

[23] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8

$$E_\epsilon(y(\mathbf{x}) - t) = \begin{cases} 0, & \text{if} |y(\mathbf{x}) - t| < \epsilon; \\ |y(\mathbf{x}) - t| - \epsilon, & \text{otherwise} \end{cases} \tag{3.10}$$

Minimizing a regularized error function given by:

$$C\sum_{n=1}^{N} E_\epsilon(y(\mathbf{x}_n) - t_n) + \frac{1}{2}\|\mathbf{w}\|^2 \tag{3.11}$$

where $E_\epsilon(y(\mathbf{x_n}) - t_n)$ is the ϵ-insensitive, function, $y(\mathbf{x}) = \mathbf{w}^T\mathbf{X} + b$ (with the bias parameter $b = w_0$ explicit) and by convention the (inverse) regularization parameter, denoted C, appears in front of the error term.

The linear support vector machine prediction function is very similar to simple linear regression model predicting new samples using linear combinations of the data and parameters. The parameter estimates can be written as functions of a set of unknown parameters (α_n) and the training set data points so that:

$$y(\mathbf{x}) = \sum_{n=1}^{N}(\alpha_n - \hat{\alpha}_n)k(\mathbf{x}, \mathbf{x}_n) + b \tag{3.12}$$

There are several aspects of this equation worth pointing out. First, there are as many α_n parameters as there are data points. From the standpoint of classical regression modeling, this model would be considered overparameterized; typically, it is better to estimate fewer parameters than data points. However, the use of the cost value effectively regularizes the model to help alleviate this problem.

Second, the individual training set data points are required for new predictions. When the training set is large, this makes the prediction equations less compact than other techniques. However, for some percentage of the training set samples, the α_n parameters will be exactly zero, indicating that they have no impact on the prediction equation. The data points associated with an α_n parameter of zero are the training set samples that are within $\pm$ of the regression line (i.e., are within the "funnel" or "tube" around the regression line (figure). As a consequence, only a subset of training set data points, where $\alpha \neq 0$, are needed for

prediction. Since the regression line is determined using these samples, they are called the support vectors as they support the regression line.

Finally, note that the new samples enter into the prediction function as sum of cross products with the new sample values. This function ($k(\mathbf{x}, \mathbf{x}_n)$) is called *kernel function*.

Types of kernel functions that can be used to generalize the regression model and encompass nonlinear functions of the predictors are:

- Linear Kernel:

$$k(\mathbf{x}, \mathbf{x}_n) = \mathbf{x} \cdot \mathbf{x}_n$$

- Polynomial Kernel:

$$k(\mathbf{x}, \mathbf{x}_n) = (\mathbf{x} \cdot \mathbf{x}_n)^d$$

$$k(\mathbf{x}, \mathbf{x}_n) = (\mathbf{x} \cdot \mathbf{x}_n + 1)^d$$

- Radial basis function Kernel:

$$k(\mathbf{x}, \mathbf{x}_n) = e^{-\frac{\|\mathbf{x}_n - \mathbf{x}_i\|}{2\sigma^2}} = e^{-\gamma \|\mathbf{x}_n - \mathbf{x}_i\|^2}$$

- Multilayer Perceptron:

$$k(\mathbf{x}, \mathbf{x}_n) = \tanh(\alpha(\mathbf{x} \cdot \mathbf{x}_n) + b$$

Where α and γ are scaling parameters and model hyperparameters.

3.6 *Decision Trees Regression*

Decision trees partition the data into smaller groups that are more homogenous with respect to the response. To achieve outcome homogeneity, regression trees determine:

- The predictor to split on and value of the split.
- The depth or complexity of the tree.
- The prediction equation in the terminal nodes.

For regression, the model begins with the entire data set, $\mathcal{D}$, and searches every distinct value of every predictor to find the predictor and split value that partitions the data into two groups ($\mathcal{D}_1$ and $\mathcal{D}_2$) such that the overall sums of squares error are minimized:

$$SSE = \sum_{n \in \mathcal{D}_1} \{y_n - t_1\}^2 + \sum_{n \in \mathcal{D}_2} \{y_n - t_2\}^2 \tag{3.13}$$

where t_1 and t_2 are the averages of the training set outcomes within groups $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. Then within each of groups $\mathcal{D}_1$ and $\mathcal{D}_2$, this method searches for the predictor and split value that best reduces SSE. Because of the recursive splitting nature of regression trees, this method is also known as recursive partitioning[24].

An advantage of tree-based models is that, when the tree is not large, the model is simple and interpretable. Also, this type of tree can be computed quickly (despite using multiple exhaustive searches). Tree models intrinsically conduct feature selection; if a predictor is never used in a split, the prediction equation is independent of these data. This advantage is weakened when there are highly correlated predictors. If two predictors are extremely correlated, the choice of which to use in a split is somewhat random. Finally, an example of a basic regression model tree is shown in figure 3.7.

[24] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

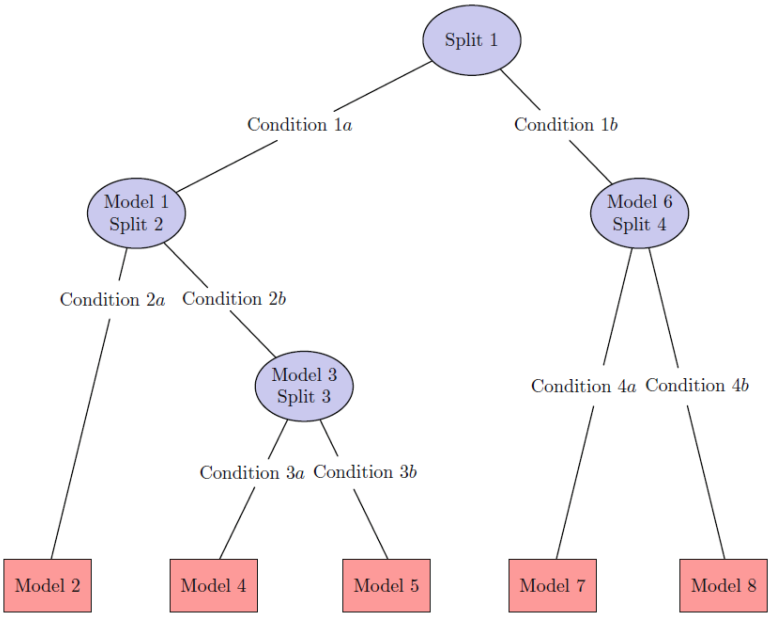

Figure 3.7: Example of a regression tree model

3.7 Bagged Tree Algorithm

Bagging, short for bootstrap aggregation,is a general approach that uses bootstrapping (sampling data with replacement, that meaning, after a data point is selected for the subset, it is still available for further selection) in conjunction with any regression model to construct an ensemble. The method is fairly simple in structure and consists of the steps in bellow algorithm:

```
1 for i = 1 to m do
2 | Generate a bootstrap sample of the original data
3 | Train an unpruned tree model on this sample
4 end
```

Each model in the ensemble is then used to generate a prediction for a new sample and these m predictions are averaged to give the bagged model's prediction.

Bagging models provide several advantages over models that are not bagged. First, bagging effectively reduces the variance of a prediction through its aggregation process. For models that produce an unstable prediction, like regression trees, aggregating over many versions of the training data actually reduces the variance in the prediction and, hence, makes the prediction more stable.When the predictions for a sample are averaged across all of the single trees, the average prediction has lower variance than the variance across the individual predictions. This means that if we were to generate a different sequence of bootstrap samples, build a model on each of the bootstrap samples, and average the predictions across models, then we would likely get a very similar predicted value for the selected sample as with the previous bagging model. This characteristic also improves the predictive performance of a bagged model over a model that is not bagged. If the goal of the modeling effort is to find the best prediction, then bagging has a distinct advantage.

3.8 Random Forest Tree Algorithm

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Generating bootstrap samples (baggingg) introduces a random component into the tree building process, which induces a distribution of trees, and therefore also a distribution of predicted values for each sample. This trees are not completely independent of each other since all of the original predictors are considered at every split of every tree. If we start with a sufficiently large number of original samples and a

relationship between predictors and response that can be adequately modeled by a tree, then trees from different bootstrap samples may have similar structures to each other due to the underlying relationship. This characteristic is known as **tree correlation** and prevents bagging from optimally reducing variance of the predicted values.

To avoid this phenomenon reducing correlation among predictors can be done by adding randomness to the tree construction process. After evaluating generalizations such as random split selection (where trees are built using a random subset of the top k predictors at each split in the tree) or build entire trees based on random subsets of descriptors to the original bagging algorithm, Breiman constructed a unified algorithm called **random forests**[25].

[25] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. DOI: 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A%3A1010933404324

A general random forests algorithm for a tree-based model can be implemented as follows:

```
Select the number of models to build, m
for i = 1 to m do
| Generate a bootstrap sample of the original data
| Train a tree model on this sample
| for each split do
| | Randomly select k(< P) of the original predictors
| | Select the best predictor among the k predictors and partition the data
| end
| Use typical tree model stopping criteria to determine when a tree is complete (but do not prune)
end
```

A random forest model achieves variance reduction by selecting strong, complex learners that exhibit low bias. Also, compared to bagging, random forests is more computationally efficient on a tree-by-tree basis since the tree building process only needs to evaluate a fraction of the original predictors at each split, although more trees are usually required by random forests[26].

[26] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

4 Methodology

Contents

4.1 Data Description **52**
4.1.1 Attribute Information 53
4.2 Data Preprocessing **53**
4.2.1 Data Profiling 53
4.2.2 Data Cleansing 54
4.2.3 Data Reduction 55
4.2.4 Data Transformation 61
4.2.5 Data Validation 62
4.3 Linear Regression Application **63**
4.3.1 All Data Linear Regression Model (ADLRM) 64
4.3.2 All Features with Raw Data Linear Regression Model (AFRLRM) 66
4.3.3 All Features with Standardized Data Linear Regression Model (AFSLRM) . 66
4.3.4 Backward Elimination with Standardized Data Linear Regression Model (BESLRM) 66
4.3.5 Stepwise Regression with Standardized Data Linear Regression Model (SRSLRM) 68
4.3.6 Operational Features with Standardized Data Linear Regression Model (OFSLRM) 68
4.3.7 Principal Component Linear Regression Model (PCALRM) 68
4.3.8 Partial Least Squares Regression Model (PLSLRM) 70
4.4 Support Vector Machine (SVM) Regression Application **70**
4.4.1 All Features SVM Regression Model: Linear kernel (AFLSVM) 71
4.4.2 All Features SVM Regression Model: RBF kernel (AFRSVM) 71

4.4.3 Feature Reduction SVM Regression Model: Linear kernel (FRLSVM) . . . 71
4.4.4 Feature Reduction SVM Regression Model: RBF kernel (FRRSVM) 71
4.5 Regression Trees Application 71
4.5.1 Decision Tree without Max Depth Defined (DNDRTM) 72
4.5.2 Decision Tree with Best Depth as Hyperparameter (DDHRTM) 72
4.5.3 Bagged Trees (BGGRTM) 72
4.5.4 Bagged Trees with Best Trees Number as Hyperparameter (BTHRTM) 73
4.5.5 Random Forest Trees (RFRRTM) . . . 73
4.5.6 Random Forest Trees with Best Trees Number as Hyperparameter (RTHRTM) 74

4.1 Data Description

Dataset is made up of information on the variables that describe the behavior of the corn gluten dryer owned by ALMEX and whose objective is to obtain product with a target **moisture**. The input variables were selected based on an operational analysis of the dewatering sub-processes (stage prior to the dryer) and drying, carried out among the process engineering and production teams of this organization based on the operational experience obtained throughout years.

Data Source: Real time Data from lab results and gluten dewatering and drying process instrumentation (as seen on figure 4.1).

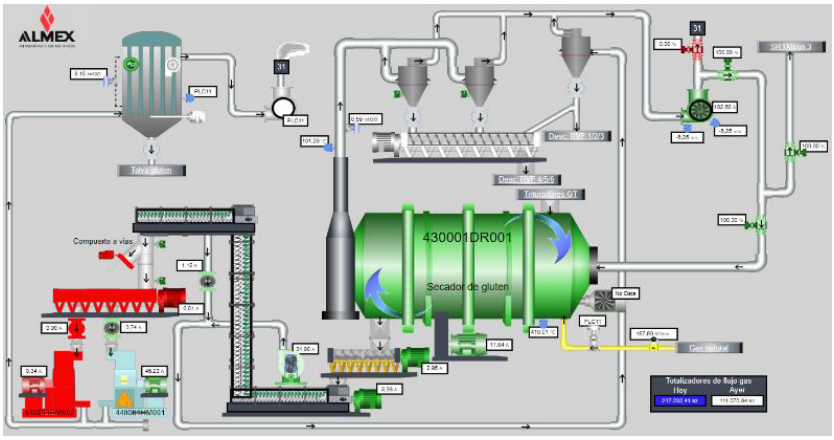

Figure 4.1: Gluten dryer real-time data from process sensors.

Dataset has **3417 instances, 34 entries and one output** attribute, being the **moisture** output attribute the reference of the time in which

the data from the measurable attributes with online instrumentation of the process should be taken. The output attribute is the result of a laboratory analysis carried out by the quality department and registered in a web page that serves as a reporter, the input attributes are obtained directly from the server of the historian software (which collect data from instrumentation sources).

4.1.1 Attribute Information

Target: (see Table 4.1).

Index	Target	Description	UOM	Zero	Span
0	Moisture	Quantity of water held in corn total mass	%	0	100

Table 4.1: Target information, such as feature description, scale and unit of measure

Predictors: (see Table 4.2).

4.2 Data Preprocessing

The need for data preprocessing is determined by the type of model being used. Some procedures, such as tree-based models, are notably insensitive to the characteristics of the predictor data. Others, like linear regression, are not[1]. Said this, our Dataset needs to pass through data preprocessing procedures to increase selected machine learning models predictive ability.

[1] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

This procedures include:

- Analysing Dataset quality (Data profiling).
- Cleaning missing values and outliers
- Create train and test subset to evaluate each model
- Center and normalize data to avoid scale problems
- Feature selection with unsupervised and supervised procedures.

All this based on Dataset quality report.

4.2.1 Data Profiling

Data Profiling is the process of examining, analyzing, reviewing and summarizing data sets to gain insight into the quality of data. This can be done through to quality report that summarizes critical information such as data type, missing values quantity, Unique values, minimum and maximum values, etc. Table 4.3 shows results for this report done via python algorithm.

4.2.2 *Data Cleansing*

Before a missing data analysis is performed, Data type must be consistent. From Data quality report (DQR) we know that 4 variables are not numerical (Object type) and can be used for the prediction models. Object type variables are operational states, so transforming to numerical type must be done. A common industry standard is 0 for *Closed* and *Stopped or Inactive* state, 1 for *Open* and *Running or Active*.

After transforming object type predictors, firs step for data preprocessing is data cleansing handling missing values. DQR shows missing values quantity for each feature it is shown as a heatmap on figure 4.2 .

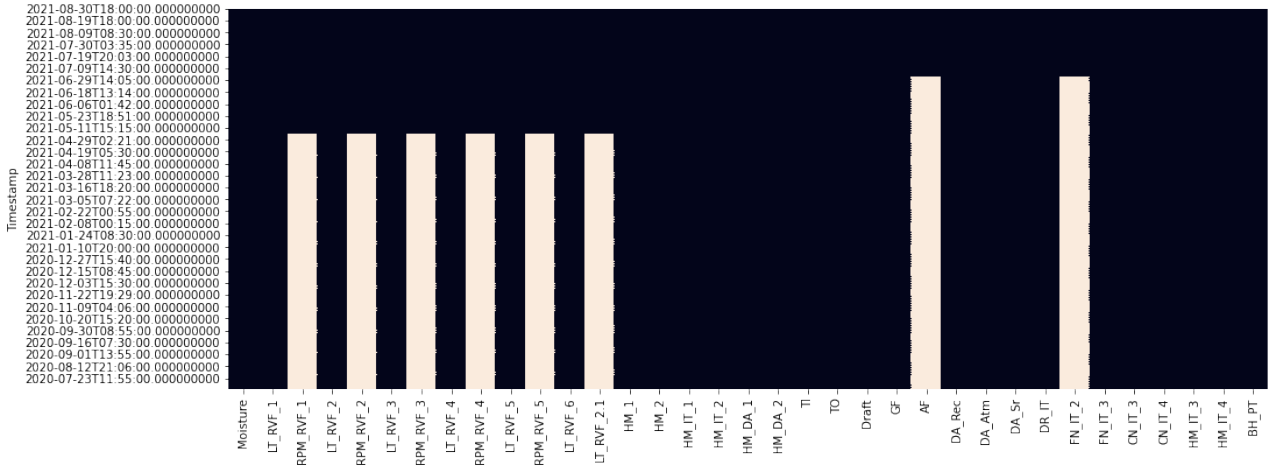

Figure 4.2: Missing Data Heatmap for Gluten Drying Process Dataset.

Prior taking decisions on how to handle data, it is necessary to know why are we having missing values. According to Rubin[2], there are three mechanisms under which missing data can occur: missing at random (MAR), missing completely at random (MCAR), and missing not at random (MNAR). In MCAR, the probability of data being missing is the same for all the observations. In this case, there is no relationship between the missing data and any other values observed or unobserved (the data which is not recorded) within the given dataset. Missing at random (MAR) means that the reason for missing values can be explained by variables on which you have complete information as there is some relationship between the missing data and other values/data. In this case, the data is not missing for all the observations. It is missing only within sub-samples of the data and there is some pattern in the missing values. Finally, in MNAR missing values depend on the unobserved data. If there is some structure/pattern in missing data and other observed data can not explain it, then it is Missing Not At Random. If the missing data does not fall under the MCAR or MAR then it can be categorized as MNAR.

[2] Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. ISSN 00063444. URL http://www.jstor.org/stable/2335739

Predictors with missing values are **RPM_RVF_1**, **RPM_RVF_2**, **RPM_RVF_3**, **RPM_RVF_4**, **RPM_RVF_5**, **RPM_RVF_6**, **AF** and **FN_IT_2**. Asking to Almex maintenance, operation and engineering teams why this predictors may have data missing the common answer was:

- Instrument failure: MCAR
- Operation not being aware to have this failure: MNAR
- Instrument is connected to old infrastructure that does not historize information: MCAR
- Variable not need it to drying operation process (just on dewatering): MNAR
- Instrument recently installed: MAR

Missing data ratio is telling us that this predictors must be discarded before thinking on an imputation model to filling data, but, as we can see from missing data analysis, all tree causes of missing data are presented on our Dataset, therefore, discard predictors just for missing data is not an option. Said this, predictors will be analyzed on data reduction section.

About outliers, regression model to identify them will be used. This will be discussed on section 4.3.4.

4.2.3 *Data Reduction*

Data reduction techniques reduce the data by generating a smaller set of predictors that seek to capture a majority of the information in the original variables. In this way, fewer variables can be used that provide reasonable fidelity to the original data. Said this, as we observed on section 4.2.2, **RPM_RVF_1**, **RPM_RVF_2**, **RPM_RVF_3**, **RPM_RVF_4**, **RPM_RVF_5**, **RPM_RVF_6**, **AF** and **FN_IT_2** predictors are candidates to be eliminated. To justify this, a subset without missing data is created in order to analyze if this variable have value info to predict our target or is correlated with another predictor with no missing data. Table 4.4 shows DQR results for this subset.

In this subset **DA_Atm** & **DA_Atm** shows only one value and by variance criterion (zero variance predictor, uninformative variable that have little effect on the calculations)[3] will be dropped for correlation analysis. This will be done through a HeatMap.

On Correlation Heatmap (figure 4.3) **FN_IT_2** is the feature with more "Dark Areas" (inverse correlation). It is observed that is correlated to **TI**, **GF** and **TO**, also is low correlated with target variable, so dropping out is valid. Equally, **AF** is the feature with more "light

[3] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

Areas" (direct correlation) correlated with **FN_IT_2** (inverse), **TI**, **GF** and **TO** justifying eliminating it. In addition, from correlation Heatmap, any **RPM_RVF** (speed) do not have any important information (not correlated to anyone) and Dataset DQR shows that has big ratio of missing values, giving us enough evidence to justify eliminating all **RPM_RVF**.

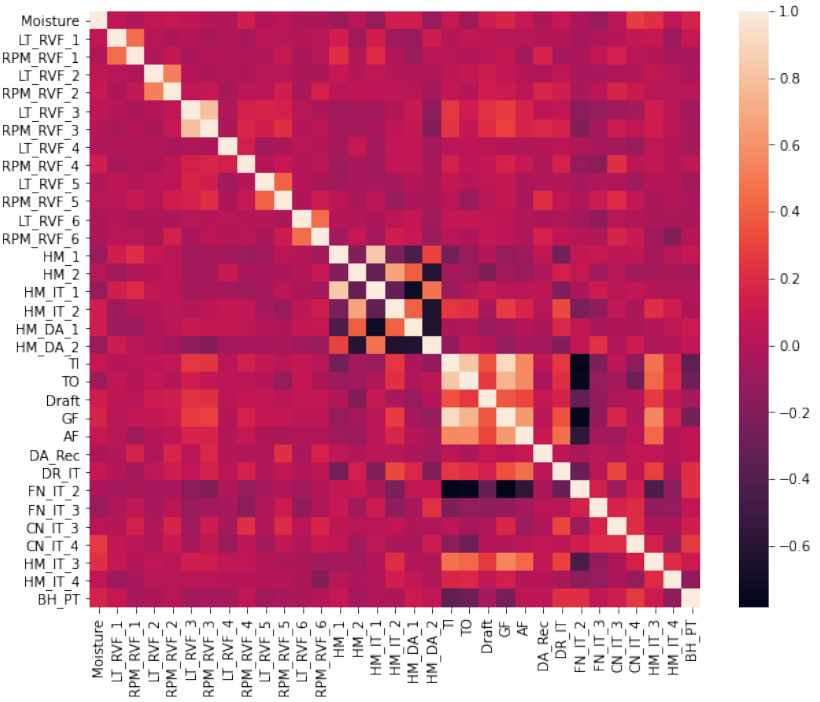

Figure 4.3: Correlation Heatmap for gluten drying process subset.

Table 4.2: Attributes information, such as feature description, scale and unit of measure

Index	Attribute	Description	UOM	Zero	Span
0	LT_RVF_1	Gluten Rotary Vacuum Filter Vat Level 1	%	0	100
1	RPM_RVF_1	Gluten Rotary Vacuum Filter Motor Speed 1	rpm	0	1800
2	LT_RVF_2	Gluten Rotary Vacuum Filter Vat Level 2	%	0	100
3	RPM_RVF_2	Gluten Rotary Vacuum Filter Motor Speed 2	rpm	0	1800
4	LT_RVF_3	Gluten Rotary Vacuum Filter Vat Level 3	%	0	100
5	RPM_RVF_3	Gluten Rotary Vacuum Filter Motor Speed 3	rpm	0	1800
6	LT_RVF_4	Gluten Rotary Vacuum Filter Vat Level 4	%	0	100
7	RPM_RVF_4	Gluten Rotary Vacuum Filter Motor Speed 4	rpm	0	1800
8	LT_RVF_5	Gluten Rotary Vacuum Filter Vat Level 5	%	0	100
9	RPM_RVF_5	Gluten Rotary Vacuum Filter Motor Speed 5	rpm	0	1800
10	LT_RVF_6	Gluten Rotary Vacuum Filter Vat Level 6	%	0	100
11	LT_RVF_2	Gluten Rotary Vacuum Filter Motor Speed 6	rpm	0	1800
12	HM_1	Gluten Feed Crusher Status 1	NA	Inactive	Active
13	HM_2	Gluten Feed Crusher Status 2	NA	Inactive	Active
14	HM_IT_1	Gluten Feed Crusher Motor Current 1	A	0	150
15	HM_IT_2	Gluten Feed Crusher Motor Current 2	A	0	150
16	HM_DA_1	Gluten Feed Crusher Bypass Damper Status 1	NA	Closed	Open
17	HM_DA_2	Gluten Feed Crusher Bypass Damper Status 2	NA	Closed	Open
18	TI	Gluten dryer Inlet Air temperature	°C	0	1000
19	TO	Gluten dryer Oulet Air temperature	°C	0	200
20	Draft	Gluten dryer Differential pressure of the air stream (Draft)	inH2O	-10	10
21	GF	Fuel Gas Flow to Gluten Dryer Burner	scfm	0	1000
22	AF	Combustion Air Flow to Gluten Dryer Burner	%	0	100
23	DA_Rec	Exhaust recycling to gluten dryer damper opening	%	0	100
24	DA_Atm	Exhaust to athmosphere from gluten dryer damper opening	%	0	100
25	DA_Sr	Exhaust to scrubber from gluten dryer damper opening	%	0	100
26	DR_IT	Gluten Dryer Drum Motor Current	A	0	50
27	FN_IT_2	Exhaust Fan motor current	A	0	180
28	FN_IT_3	Recycle Blower Motor Current	A	0	50
29	CN_IT_3	Gluten Dryer Discharge Screw conveyor Motor Current	A	0	10
30	CN_IT_4	Gluten Dryer Discharge Conveyor Motor Current	A	0	20
31	HM_IT_3	Dry gluten mill motor current 1	A	0	250
32	HM_IT_4	Dry gluten mill motor current 2	A	0	250
33	BH_PT	Dry gluten conveying bag filter differential pressure	inH2O	-10	10

Feature	Type	Missing	Present	Unique	Min	Max
Moisture	float64	0	3416	894	1.57	17.77
LT_RVF_1	float64	0	3416	3410	0.264	103.64
RPM_RVF_1	float64	2283	1133	1071	0	959.412
LT_RVF_2	float64	0	3416	3270	0	91.377
RPM_RVF_2	float64	2283	1133	1052	0	1799.998
LT_RVF_3	float64	0	3416	3320	0	100
RPM_RVF_3	float64	2283	1133	1118	0	1800
LT_RVF_4	float64	0	3416	3339	0	100
RPM_RVF_4	float64	2283	1133	1122	0.02	1800
LT_RVF_5	float64	0	3416	3195	1.835	103.64
RPM_RVF_5	float64	2283	1133	1093	0	1798.505
LT_RVF_6	float64	0	3416	3128	2.326	101.073
LT_RVF_2.1	float64	2283	1133	1087	0	1786.765
HM_1	object	0	3416	2	Active	Inactive
HM_2	object	0	3416	2	Active	Inactive
HM_IT_1	float64	0	3416	2413	0	388.579
HM_IT_2	float64	0	3416	1246	0	121.134
HM_DA_1	object	0	3416	2	Closed	Open
HM_DA_2	object	0	3416	2	Closed	Open
TI	float64	0	3416	3280	110.942	574.014
TO	float64	0	3416	3265	27.905	134.805
Draft	float64	0	3416	3415	-1.269	1.375
GF	float64	0	3416	3409	0	337.808
AF	float64	2799	617	599	4.561	9.8
DA_Rec	float64	0	3416	48	0	100
DA_Atm	float64	0	3416	8	0	100
DA_Sr	int64	0	3416	2	0	100
DR_IT	float64	0	3416	3414	0.05	26.749
FN_IT_2	float64	2799	617	599	100.511	124.562
FN_IT_3	float64	0	3416	3414	0.051	33.701
CN_IT_3	float64	0	3416	3411	0.006	3.178
CN_IT_4	float64	0	3416	3413	-0.019	4.594
HM_IT_3	float64	0	3416	3404	0.009	168.607
HM_IT_4	float64	0	3416	2972	0.319	139.583
BH_PT	float64	0	3416	3413	-0.032	6.487

Table 4.3: Data quality report for Gluten Drying Process Dataset.

Table 4.4: Data quality report for Gluten Drying Process subset

Feature	Type	Missing	Present	Unique	Min	Max
Moisture	float64	0	617	393	4.27	16.14
LT_RVF_1	float64	0	617	617	7.097	93.951
RPM_RVF_1	float64	0	617	591	0	709.197
LT_RVF_2	float64	0	617	603	0	88.984
RPM_RVF_2	float64	0	617	609	0	1096.49
LT_RVF_3	float64	0	617	607	0	89.92
RPM_RVF_3	float64	0	617	610	0	1104.478
LT_RVF_4	float64	0	617	616	0.004	100
RPM_RVF_4	float64	0	617	614	0.041	1111.126
LT_RVF_5	float64	0	617	617	2.193	103.64
RPM_RVF_5	float64	0	617	602	0	945.493
LT_RVF_6	float64	0	617	617	7.107	101.073
LT_RVF_2.1	float64	0	617	598	0	964.202
HM_1	int64	0	617	2	0	1
HM_2	int64	0	617	2	0	1
HM_IT_1	float64	0	617	412	0	98.553
HM_IT_2	float64	0	617	553	0	121.134
HM_DA_1	int64	0	617	2	0	1
HM_DA_2	int64	0	617	2	0	1
TI	float64	0	617	614	256.034	567.103
TO	float64	0	617	606	75.217	127.828
Draft	float64	0	617	617	-1.269	1.034
GF	float64	0	617	617	0	289.544
AF	float64	0	617	599	4.561	9.8
DA_Rec	float64	0	617	2	39.254	100
DA_Atm	float64	0	617	1	0	0
DA_Sr	int64	0	617	1	100	100
DR_IT	float64	0	617	617	0.072	26.749
FN_IT_2	float64	0	617	599	100.511	124.562
FN_IT_3	float64	0	617	617	15.672	29.241
CN_IT_3	float64	0	617	617	0.007	3.093
CN_IT_4	float64	0	617	617	2.759	3.792
HM_IT_3	float64	0	617	617	43.518	133.404
HM_IT_4	float64	0	617	551	0.323	44.212
BH_PT	float64	0	617	617	0.963	1.948

Finally, correlation between **GF** and **TI** is near 0.89 (strong light area) and Figure 4.4 confirms it, thus, **GF** is eliminated (**TI** is considerate a *critical operation variable* in Gluten Dryer operation[4]).

Table 4.5 presents DQR result for clean Dataset , and figure 4.5 confirm it graphically.

[4] Paul Harwood Blanchard. *Technology of Corn Wet Milling and Associated Processes Volume 4 of Industrial chemistry library*. Elsevier, first edition, May 1992. ISBN 0-4448825-5-3

Table 4.5: Data quality report for clean Gluten Drying Process

Feature	Type	Missing	Present	Unique	Min	Max
Moisture	float64	0	3416	894	1.57	17.77
LT_RVF_1	float64	0	3416	3410	0.264	103.64
LT_RVF_2	float64	0	3416	3270	0	91.377
LT_RVF_3	float64	0	3416	3320	0	100
LT_RVF_4	float64	0	3416	3339	0	100
LT_RVF_5	float64	0	3416	3195	1.835	103.64
LT_RVF_6	float64	0	3416	3128	2.326	101.073
HM_1	int64	0	3416	2	0	1
HM_2	int64	0	3416	2	0	1
HM_IT_1	float64	0	3416	2413	0	388.579
HM_IT_2	float64	0	3416	1246	0	121.134
HM_DA_1	int64	0	3416	2	0	1
HM_DA_2	int64	0	3416	2	0	1
TI	float64	0	3416	3280	110.942	574.014
TO	float64	0	3416	3265	27.905	134.805
Draft	float64	0	3416	3415	-1.269	1.375
DA_Rec	float64	0	3416	48	0	100
DA_Atm	float64	0	3416	8	0	100
DA_Sr	int64	0	3416	2	0	100
DR_IT	float64	0	3416	3414	0.05	26.749
FN_IT_3	float64	0	3416	3414	0.051	33.701
CN_IT_3	float64	0	3416	3411	0.006	3.178
CN_IT_4	float64	0	3416	3413	-0.019	4.594
HM_IT_3	float64	0	3416	3404	0.009	168.607
HM_IT_4	float64	0	3416	2972	0.319	139.583
BH_PT	float64	0	3416	3413	-0.032	6.487

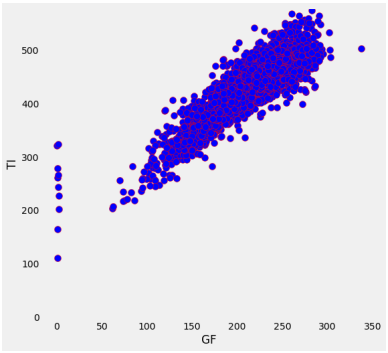

Figure 4.4: Correlation between GF & TI.

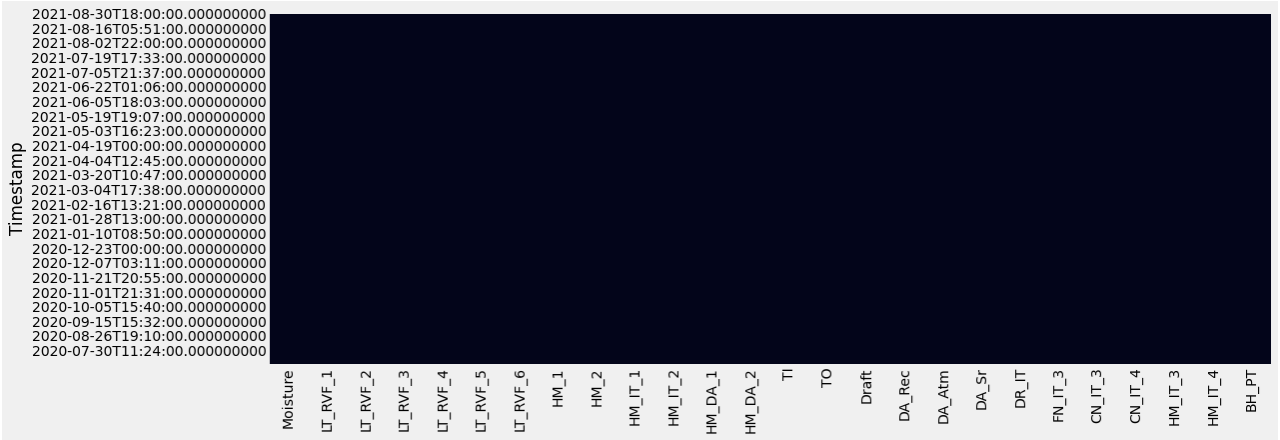

Figure 4.5: Missing Data Heatmap for Clean Gluten Drying Process Dataset.

4.2.4 *Data Transformation*

We know that data transformation refers to process of changing the format, structure, or values of data. Said this, on 4.2.2 we did a first transformation to change data type *Object* to *Float* in order to have consistent data through Dataset.

Another common reason for transformations is to remove distributional skewness. Figure 4.6 shows each feature distribution.

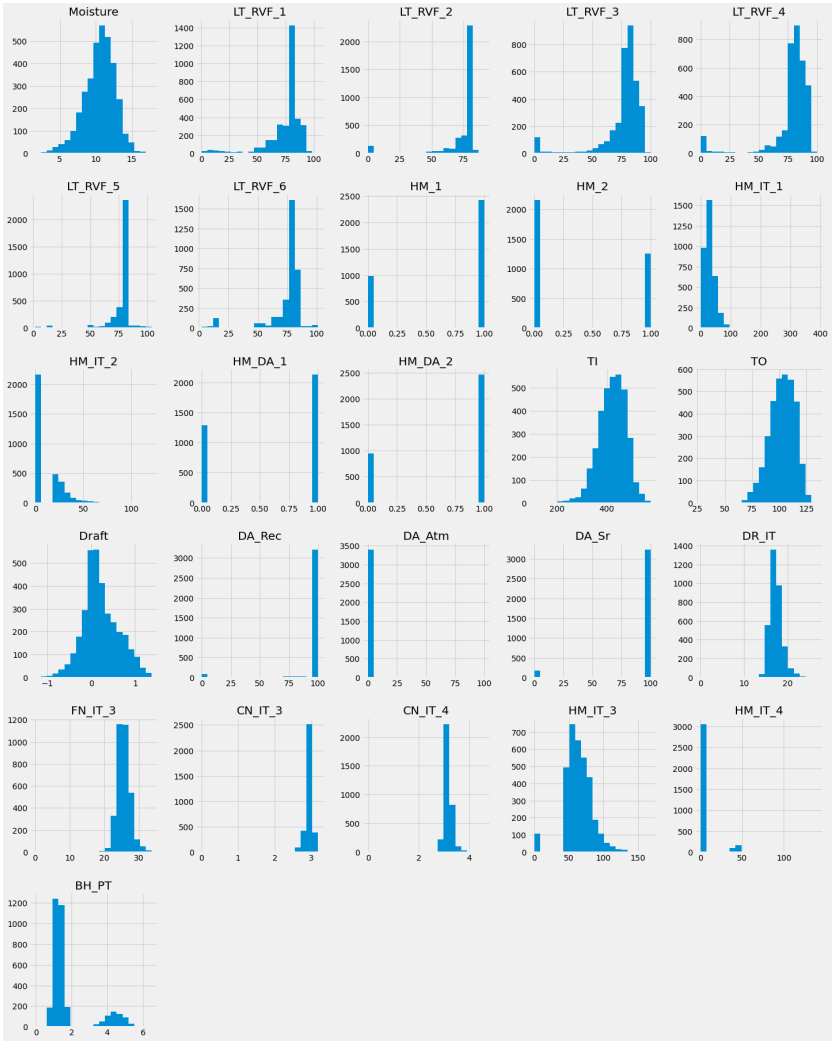

Figure 4.6: Correlation Heatmap for clean Gluten Drying Process Dataset.

A general rule of thumb to consider is that skewed data whose ratio of the highest value to the lowest value is greater than 20 have significant skewness[5].However, for each predictor kurtosis has an empirical explanation and is important to conserve it to understand how this affect **Moisture** prediction. On table 4.6 a cause of skewness in each predictor is explained. Said this, variables will not be transformed

[5] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

to eliminate skewness.

Feature	Cause of skewness
LT_RVF_1	Operational or maintenance shutdown, vat is emptied.
LT_RVF_2	Operational or maintenance shutdown, vat is emptied.
LT_RVF_3	Operational or maintenance shutdown, vat is emptied.
LT_RVF_4	Operational or maintenance shutdown, vat is emptied.
LT_RVF_5	Operational or maintenance shutdown, vat is emptied.
LT_RVF_6	Operational or maintenance shutdown, vat is emptied.
HM_1	Shutdown and operation values.
HM_2	Shutdown and operation values.
HM_IT_1	Span values due equipment start, zero values due equipment shutdown.
HM_IT_2	Span values due equipment start, zero values due equipment shutdown.
HM_DA_1	Open and Closed values.
HM_DA_2	Open and Closed values.
TI	Zero values due equipment shutdown.
TO	Zero values due equipment shutdown.
Draft	Zero values due equipment shutdown.
DA_Rec	Zero values due equipment shutdown.
DA_Atm	Zero values due equipment shutdown.
DA_Sr	Zero values due equipment shutdown.
DR_IT	Zero values due equipment shutdown.
FN_IT_3	Span values due equipment start, zero values due equipment shutdown.
CN_IT_3	Span values due equipment start, zero values due equipment shutdown.
CN_IT_4	Span values due equipment start, zero values due equipment shutdown.
HM_IT_3	Span values due equipment start, zero values due equipment shutdown.
HM_IT_4	Span values due equipment start, zero values due equipment shutdown.
BH_PT	Bimodal distribution due operation, 0-2 range normal operation, 3-6 range filter is saturated.

Table 4.6: Cause of skewness of each variable

Finally, clean dataset DQR (Table 4.5) shows that predictors has considerable range difference, therefore, centering and scaling will be performed on prediction models that are sensible to this. In order to compare multiple models performance, two datasets will be used: one standardized (centered and scaled) and one "raw". Target variable remains in its original scale to not loose interpretability. Refer to section 4.2.5 to view standardized Dataset DQR posterior to Data Validation procedures (Train and Test split).

4.2.5 *Data Validation*

One of the first decisions to make when modeling is to decide which samples will be used to evaluate performance. When a large amount of data is at hand, a set of samples can be set aside to evaluate the final model. The *training* data set is the general term for the samples used to create the model, while the *test* data set is used to qualify performance. In most cases, there is the desire to make the training and test sets as homogeneous as possible. Random sampling methods can be used to create similar data sets.[6]

For our prediction models, two subsets will be created, 80% of data to train and 20% to test it. In order to have repeatability on

[6] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

our python algorithms, will be used the 42 random state as a normal standard on ML algorithms. This is based on a pop-culture reference in Douglas Adams's popular science-fiction novel *The Hitchhiker's Guide to the Galaxy*, towards the end of the book, the supercomputer Deep Thought reveals that the answer to the great question of "life, the universe and everything" is 42[7]. Python library to perform this split is Sklearn[8].

Table 4.7, Table 4.8, Table 4.9 and Table 4.10 shows DQR for Train and Test subsets respectively, both for "Raw" and standardized data as was mentioned on section 4.2.4.

[7] Douglas Adams. The hitchhiker's guide to the galaxy. 1995

[8] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 01 2012

Table 4.7: Data quality report for train "Raw" Gluten Drying Process subset

Feature	Type	Missing	Present	Unique	Min	Max
Moisture	float64	0	2728	854	1.57	17.77
LT_RVF_1	float64	0	2728	2723	0.264	103.64
LT_RVF_2	float64	0	2728	2618	0	91.377
LT_RVF_3	float64	0	2728	2656	0	100
LT_RVF_4	float64	0	2728	2666	0	100
LT_RVF_5	float64	0	2728	2552	1.835	103.64
LT_RVF_6	float64	0	2728	2497	2.326	101.073
HM_1	int64	0	2728	2	0	1
HM_2	int64	0	2728	2	0	1
HM_IT_1	float64	0	2728	1938	0	315.863
HM_IT_2	float64	0	2728	967	0	121.134
HM_DA_1	int64	0	2728	2	0	1
HM_DA_2	int64	0	2728	2	0	1
TI	float64	0	2728	2640	202.382	574.014
TO	float64	0	2728	2620	65.696	134.805
Draft	float64	0	2728	2727	-1.269	1.375
DA_Rec	float64	0	2728	45	0	100
DA_Atm	float64	0	2728	4	0	25
DA_Sr	int64	0	2728	2	0	100
DR_IT	float64	0	2728	2726	13.191	25.811
FN_IT_3	float64	0	2728	2727	15.06	33.701
CN_IT_3	float64	0	2728	2726	2.525	3.178
CN_IT_4	float64	0	2728	2725	2.771	4.594
HM_IT_3	float64	0	2728	2720	0.009	168.607
HM_IT_4	float64	0	2728	2377	0.321	139.583
BH_PT	float64	0	2728	2726	0.09	6.487

4.3 *Linear Regression Application*

Several multiple linear regression models are used to predict target variable **moisture**. This are selected as a start because are the simplest prediction model. Linear models have follow considerations:

- For all multiple linear regression models is used the OLS (ordinary least squares) estimator for parameters obtaining.
- Python library Statsmodel[9] is used to estimate parameters.
- Statsmodel OLS linear regression library needs a constant attribute (a 1's vector), and will be added as "const" predictor.

[9] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. *Proceedings of the 9th Python in Science Conference*, 2010, 01 2010

Feature	Type	Missing	Present	Unique	Min	Max
Moisture	float64	0	682	442	2.42	16.57
LT_RVF_1	float64	0	682	682	4.567	100.426
LT_RVF_2	float64	0	682	653	0	88.931
LT_RVF_3	float64	0	682	665	0	96.182
LT_RVF_4	float64	0	682	673	0	99.758
LT_RVF_5	float64	0	682	658	2.292	100
LT_RVF_6	float64	0	682	645	3.636	99.244
HM_1	int64	0	682	2	0	1
HM_2	int64	0	682	2	0	1
HM_IT_1	float64	0	682	475	0	102.899
HM_IT_2	float64	0	682	279	0	87.136
HM_DA_1	int64	0	682	2	0	1
HM_DA_2	int64	0	682	2	0	1
TI	float64	0	682	674	235.553	558.631
TO	float64	0	682	676	65.446	127.541
Draft	float64	0	682	682	-1.195	1.288
DA_Rec	float64	0	682	13	0	100
DA_Atm	float64	0	682	5	0	40
DA_Sr	int64	0	682	2	0	100
DR_IT	float64	0	682	682	0.068	26.749
FN_IT_3	float64	0	682	682	14.483	33.416
CN_IT_3	float64	0	682	682	2.503	3.153
CN_IT_4	float64	0	682	682	2.677	4.232
HM_IT_3	float64	0	682	679	0.071	130.581
HM_IT_4	float64	0	682	604	0.319	127.28
BH_PT	float64	0	682	682	0.126	6.11

Table 4.8: Data quality report for test "Raw" Gluten Drying Process subset

- Model is fitted without regularization.

4.3.1 *All Data Linear Regression Model (ADLRM)*

First linear model is with no feature selection, no train data and no standardized data. It is decided to use linear regression analysis techniques to identify abnormal (outliers) data as was mentioned on section 4.2.2.

On figure 4.7 it is seen that instances 2232, 2086, 537, 3173, 2961 and 2818 are outliers. Analyzing point by point sample errors are the main cause of this outliers (we can not have any low moisture data if the dryer is shutdown). This samples will be eliminated.

As was mentioned on section 3.4.1, *ordinary least squares* parameter estimation method needs to present homoscedasticity. In figure 4.8, residuals histogram residuals seems to be normally distributed. Figure 4.9 shows a QQplot that seems to confirm past assertion. Finally, a Shapiro-Wilk test is performed. The Shapiro-Wilk test is a hypothesis test that is applied to a sample and whose null hypothesis is that the sample has been generated from a normal distribution. If the p-value is low, we can reject such a null hypothesis and say that the sample has not been generated from a normal distribution[10].

With a p-value bigger than our significance level $\alpha = 0.05$ ($0.2155 > 0.05$), we can confirm that our OLS models are homoscedastic.

[10] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4): 591–611, 1965. ISSN 00063444. URL http://www.jstor.org/stable/2333709

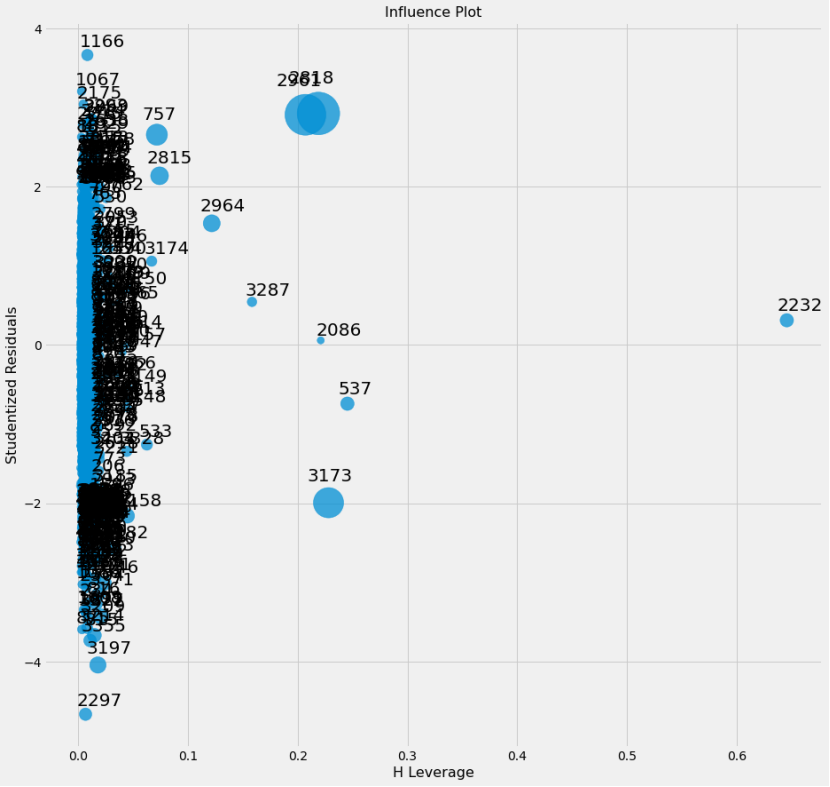

Figure 4.7: Influence plot for Gluten Drying Process dataset.

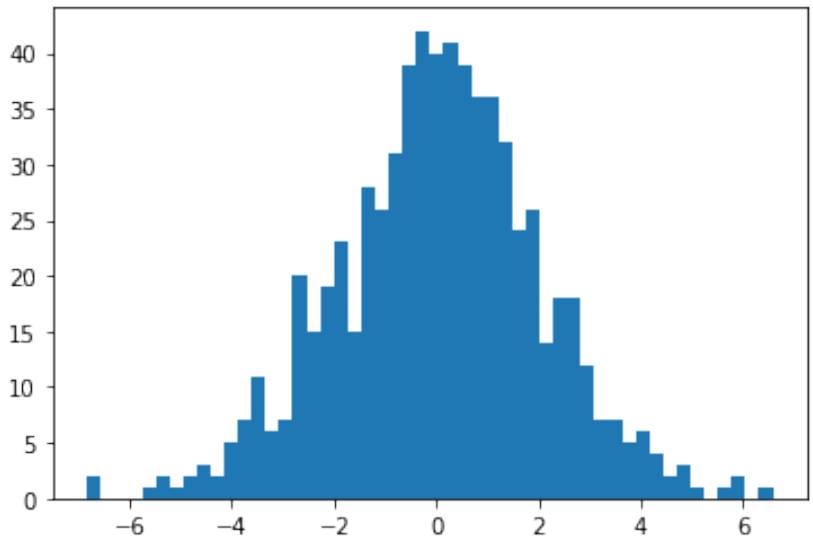

Figure 4.8: All Data Linear Regression Model residuals histogram

Feature	Type	Missing	Present	Unique	Min	Max
Moisture	float64	0	2728	854	1.57	17.77
LT_RVF_1	float64	0	2728	2723	-4.48	1.774
LT_RVF_2	float64	0	2728	2618	-4.341	1.064
LT_RVF_3	float64	0	2728	2656	-4.08	1.292
LT_RVF_4	float64	0	2728	2666	-4.163	1.198
LT_RVF_5	float64	0	2728	2552	-5.987	2.141
LT_RVF_6	float64	0	2728	2497	-4.351	1.636
HM_1	float64	0	2728	2	-1.572	0.636
HM_2	float64	0	2728	2	-0.764	1.309
HM_IT_1	float64	0	2728	1938	-1.241	13.348
HM_IT_2	float64	0	2728	967	-0.673	6.69
HM_DA_1	float64	0	2728	2	-1.284	0.779
HM_DA_2	float64	0	2728	2	-1.61	0.621
TI	float64	0	2728	2640	-4.066	2.833
TO	float64	0	2728	2620	-3.247	2.807
Draft	float64	0	2728	2727	-3.664	2.863
DA_Rec	float64	0	2728	45	-6.104	0.202
DA_Atm	float64	0	2728	4	-0.054	18.172
DA_Sr	float64	0	2728	2	-4.326	0.231
DR_IT	float64	0	2728	2726	-2.763	5.659
FN_IT_3	float64	0	2728	2727	-5.716	4.36
CN_IT_3	float64	0	2728	2726	-4.715	2.854
CN_IT_4	float64	0	2728	2725	-2.493	9.162
HM_IT_3	float64	0	2728	2720	-3.253	5.204
HM_IT_4	float64	0	2728	2377	-0.323	7.741
BH_PT	float64	0	2728	2726	-1.393	3.755

Table 4.9: Data quality report for train standardized Gluten Drying Process subset

4.3.2 *All Features with Raw Data Linear Regression Model (AFRLRM)*

Second linear model is with no feature selection, no train data and no standardized data. It is decided to take this model as a baseline because is the one with most of predictors. Results of Multiple Linear Regression Models like this one will be presented on section 5.1.1.

4.3.3 *All Features with Standardized Data Linear Regression Model (AFSLRM)*

Third model is with no feature selection, and standardized data. Results of Multiple Linear Regression Models like this one will be presented on section 5.1.1.

4.3.4 *Backward Elimination with Standardized Data Linear Regression Model (BESLRM)*

Fourth model is with Backward elimination feature selection and standardized data. It is decided to use only significant predictors with a $\alpha = 0.05$[11] significance level. Table 4.11 shows features obtained by backward elimination method and final p-value. Results of Multiple Linear Regression Models like this one will be presented on section 5.1.1.

[11] Norman R. Draper and Harry Smith. *Applied Regression Analysis*. Wiley-Interscience, third edition, 1998. ISBN 0-471-17082-8

Feature	Type	Missing	Present	Unique	Min	Max
Moisture	float64	0	682	442	2.42	16.57
LT_RVF_1	float64	0	682	682	-4.219	1.58
LT_RVF_2	float64	0	682	653	-4.341	0.919
LT_RVF_3	float64	0	682	665	-4.08	1.087
LT_RVF_4	float64	0	682	673	-4.163	1.185
LT_RVF_5	float64	0	682	658	-5.951	1.85
LT_RVF_6	float64	0	682	645	-4.272	1.525
HM_1	float64	0	682	2	-1.572	0.636
HM_2	float64	0	682	2	-0.764	1.309
HM_IT_1	float64	0	682	475	-1.241	3.511
HM_IT_2	float64	0	682	279	-0.673	4.623
HM_DA_1	float64	0	682	2	-1.284	0.779
HM_DA_2	float64	0	682	2	-1.61	0.621
TI	float64	0	682	674	-3.45	2.548
TO	float64	0	682	676	-3.269	2.171
Draft	float64	0	682	682	-3.482	2.647
DA_Rec	float64	0	682	13	-6.104	0.202
DA_Atm	float64	0	682	5	-0.054	29.107
DA_Sr	float64	0	682	2	-4.326	0.231
DR_IT	float64	0	682	682	-11.521	6.285
FN_IT_3	float64	0	682	682	-6.028	4.206
CN_IT_3	float64	0	682	682	-4.97	2.566
CN_IT_4	float64	0	682	682	-3.095	6.847
HM_IT_3	float64	0	682	679	-3.25	3.297
HM_IT_4	float64	0	682	604	-0.323	7.029
BH_PT	float64	0	682	682	-1.364	3.452

Table 4.10: Data quality report for test standardized Gluten Drying Process subset

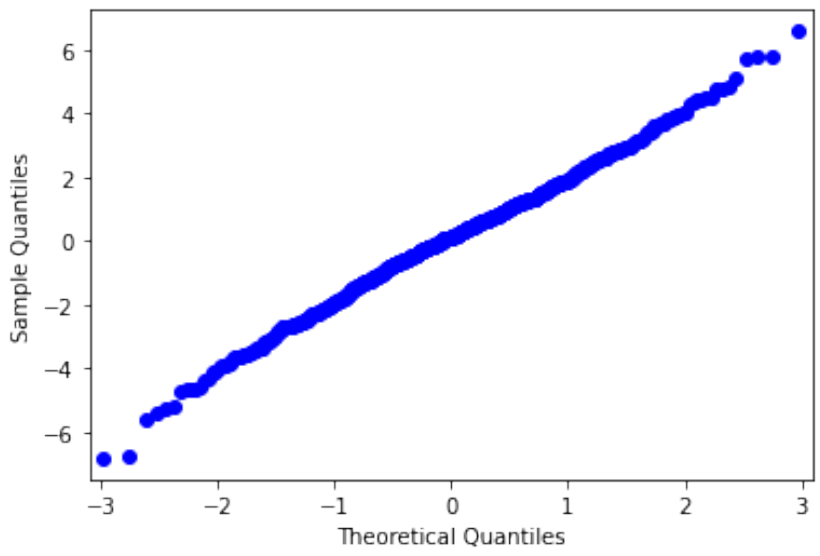

Figure 4.9: All Data Linear Regression Model residuals qqplot

4.3.5 *Stepwise Regression with Standardized Data Linear Regression Model (SRSLRM)*

Fifth model is with Stepwise elimination feature selection and standardized data. It is decided to use a feature engineering algorithm to obtain significant variables. Table 4.12 shows features obtained by Stepwise Regression method and p-value. Results of Multiple Linear Regression Models like this one will be presented on section 5.1.1.

4.3.6 *Operational Features with Standardized Data Linear Regression Model (OFSLRM)*

Sixth model is with feature selection based on operational experience and standardized data. Operation usually takes special care on following features because are where heat transfer phenomenon happens[12]:

- **TI**: how much heat enters to dryer.
- **TO**: how much heat leaves the dryer.
- **Draft**: how much hot air is going through the dryer.
- **DR_IT**: as a reference of amount of corn gluten mass is going through the dryer.
- **DA_Rec**: how much hot air is recirculated to the dryer.

Results of Multiple Linear Regression Models like this one will be presented on section 5.1.1.

4.3.7 *Principal Component Linear Regression Model (PCALRM)*

Next model is a principal component regression. Principal component regression (PCR) is a regression analysis technique that is based on principal component analysis (PCA). In PCR, instead of regressing the dependent variable on the explanatory variables directly, the principal components of the explanatory variables are used as regressors.

Sklearn library is used to obtain principal components. From cumulative explained variance screeplot (figure 4.10) we can conclude that 22 principal components describe 98.4% of data variance.

In order to confirm this principal components optimal number, RMSE VS. Number of PC plot is obtained (figure 4.11) where lowest RMSE value stabilizes without being total of the principal components number is the optimal PC number (22).

Feature	p-value
LT_RVF_1	0.007
LT_RVF_4	0.028
HM_IT_1	0
HM_DA_1	0
HM_DA_2	0
TI	0
TO	0
Draft	0
DA_Rec	0.006
DA_Sr	0.001
DR_IT	0
FN_IT_3	0.007
CN_IT_3	0.006
CN_IT_4	0
HM_IT_3	0
BH_PT	0

Table 4.11: Features obtained by backward elimination method

Feature	p-value
CN_IT_4	3.86931E-84
HM_IT_3	5.562E-27
DR_IT	0.00000000156902
DA_Sr	0.000000590319
FN_IT_3	0.0000054565
CN_IT_3	0.000199821
BH_PT	0.0000170787
LT_RVF_1	0.00109121
TI	0.00466435
TO	0.0000000278795
HM_2	0.0000786434
HM_DA_1	0.00000248261
HM_IT_1	0.0000063482
Draft	0.000582507
HM_DA_2	0.00827148
HM_2	0.253713

Table 4.12: Features obtained by stepwise regression method

[12] Paul Harwood Blanchard. *Technology of Corn Wet Milling and Associated Processes Volume 4 of Industrial chemistry library*. Elsevier, first edition, May 1992. ISBN 0-44488825-5-3

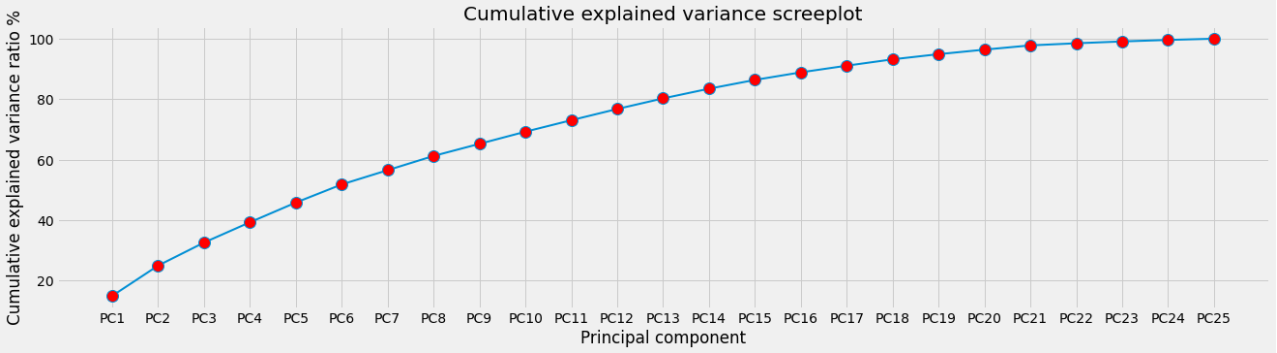

Figure 4.10: Cumulative explained variance screeplot.

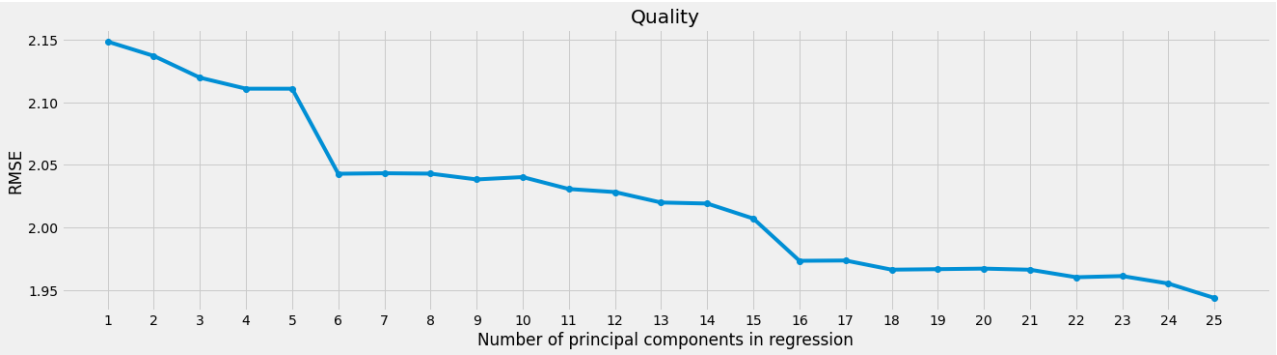

Figure 4.11: RMSE VS. Number of PC plot.

Results of Multiple Linear Regression Models like this one will be presented on section 5.1.1.

4.3.8 *Partial Least Squares Regression Model (PLSLRM)*

Final linear model is a Partial Least Squares regression. Is a quick, efficient and optimal regression method based on covariance.

This model is also obtained through SKlearn library. In order to know the optimal value of components, a cross validation algorithm with MSE is performed and results are observed on figure 4.12.

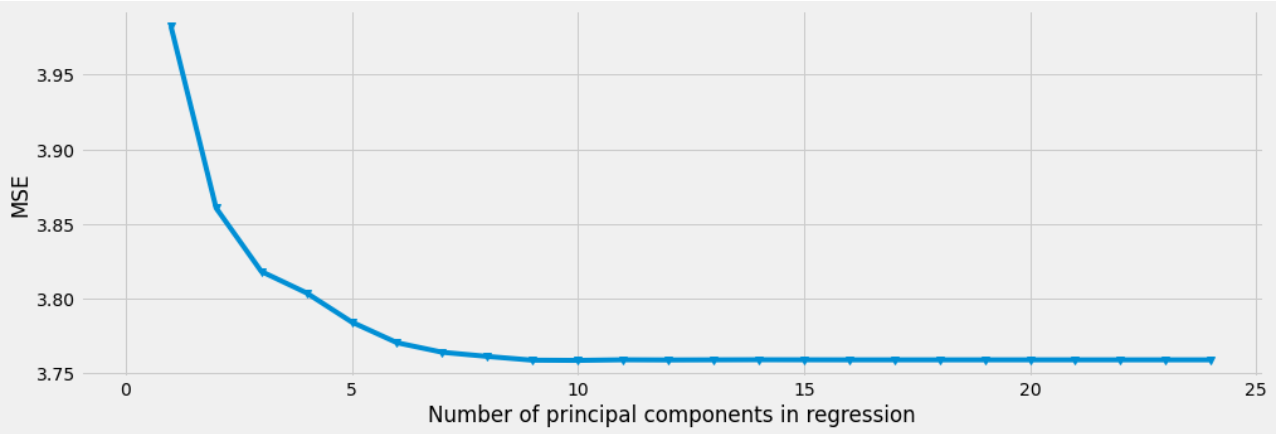

Figure 4.12: MSE VS. Number of PC regressors

At 10 components, MSE seems to stabilize and is the number of components used to estimate model parameters.Results of Multiple Linear Regression Models like this one will be presented on section 5.1.1.

4.4 *Support Vector Machine (SVM) Regression Application*

Support vector machine regression models gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data. In contrast to OLS, the objective function of SVR is to minimize the coefficients — more specifically, the l2-norm of the coefficient vector (or support vectors) — not the squared error. Said this, similar data structure to used on Multiple linear regression models will be used for SVR models, and with follow considerations:

- Python library Sklearn is used to estimate parameters.
- To compare performance, hyperparameters $\epsilon = 0.01$ and $C = 1$ will be the same on each model based on Kuhn[13] examples, this work

[13] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

only will be focused on performance based on predictors quantity and kernel used.

- Data is standardized for all SVMR models.

4.4.1 *All Features SVM Regression Model: Linear kernel (AFLSVM)*

First SVMR model is with all predictors for clean Gluten Drying Process dataset and using a linear kernel. Results of Support Vector Machine Regression Models like this one will be presented on section 5.1.2.

4.4.2 *All Features SVM Regression Model: RBF kernel (AFRSVM)*

On second SVMR model all predictors for clean Gluten Drying Process dataset will be used and kernel is changed to Radial basis function (RBF). Results of Support Vector Machine Regression Models like this one will be presented on section 5.1.2.

4.4.3 *Feature Reduction SVM Regression Model: Linear kernel (FRLSVM)*

In this SVMR model, predictors obtained from stepwise regression procedure in section 4.3.5 (because the best performance for multiple linear regression models were obtained) and using a linear kernel. Results of Support Vector Machine Regression Models like this one will be presented on section 5.1.2.

4.4.4 *Feature Reduction SVM Regression Model: RBF kernel (FRRSVM)*

Final SVMR model, predictors obtained from stepwise regression procedure in section 4.3.5 (because the best performance for multiple linear regression models were obtained) with "RBF" kernel. Results of Support Vector Machine Regression Models like this one will be presented on section 5.1.2.

4.5 *Regression Trees Application*

The general regression tree building methodology allows input variables to be a mixture of continuous and categorical variables. A decision tree is generated when each decision node in the tree contains a test on some input variable's value. The terminal nodes of the tree contain the predicted output variable values. A Regression tree may be considered as a variant of decision trees, designed to approximate real-valued functions, instead of being used for classification methods. For this work follow considerations for our regression tree models are:

- Python library Sklearn is used to estimate model.

- Dataset used is with "raw" Data knowing that this prediction models are not sensible to scale data or feature reduction[14].

[14] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3

4.5.1 *Decision Tree without Max Depth Defined (DNDRTM)*

First Decision Tree (DT) model is without max depth defined. The function to measure the quality of a split is the mean squared error, which is equal to variance reduction as feature selection criterion and minimizes the L2 loss using the mean of each terminal node. The strategy used to choose the split at each node is "best" to choose the best split. Minimum number of samples required to split an internal node and minimum number of samples required to be at a leaf node are defaults **2** and **1** respectively. Results of Regression Trees Models like this one will be presented on section 5.1.3.

4.5.2 *Decision Tree with Best Depth as Hyperparameter (DDHRTM)*

Decision Tree models main problem is that if a proper depth value to stop the algorithm is not defined, it will continue until variance stabilizes causing overfitting. To avoid this, we use a cross validation algorithm in order to have optimal max depth number. Figure 4.13 and figure 4.14 shows results of this algorithm with **20** as optimal max depth for tree.

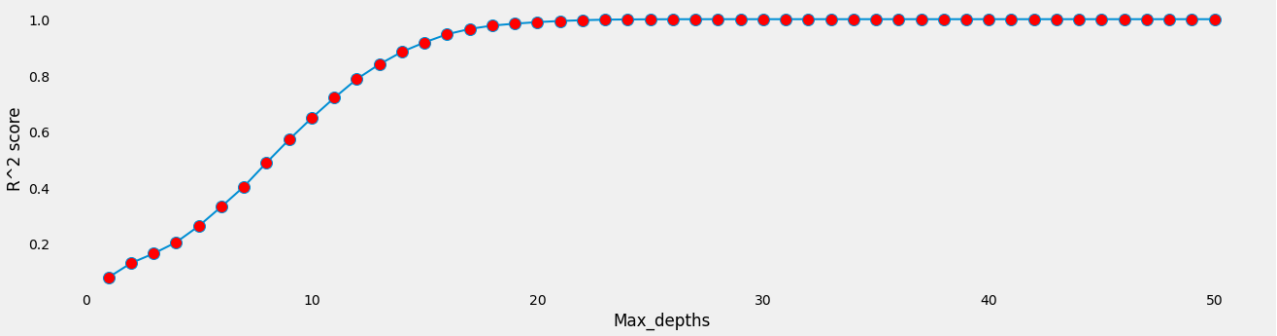

Figure 4.13: R^2 score VS. Tree max depth

Results of Regression Trees Models like this one will be presented on section 5.1.3.

4.5.3 *Bagged Trees (BGGRTM)*

When a simple decision tree is not enough, a bagging model can be used to improve results or to try to avoid overfitting. The base estimator to fit on random subsets of the dataset used is *DecisionTreeRegressor* and it's considerations. The number of base estimators in the ensemble selected is **50**. The number of samples to draw from X to train each base

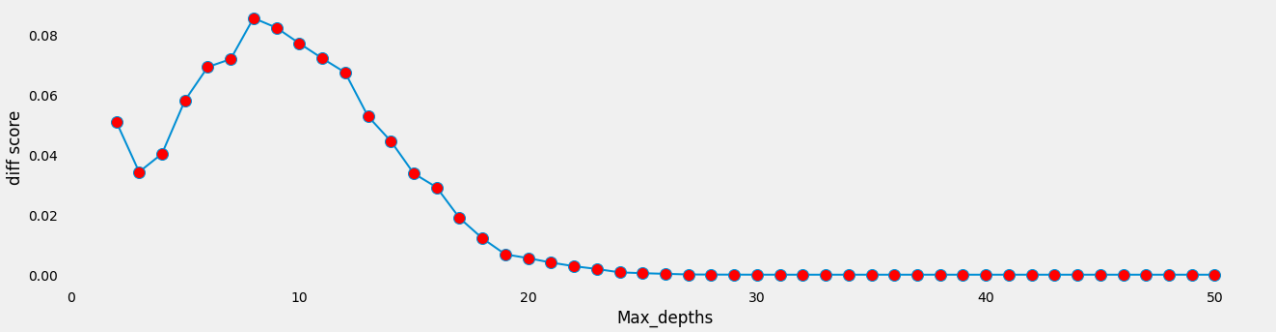

Figure 4.14: Score difference VS. Tree max depth.

estimator is **0.8** (same as train - test split ratio). Out-of-bag samples to estimate the generalization error will not be used. Results of Regression Trees Models like this one will be presented on section 5.1.3.

4.5.4 *Bagged Trees with Best Trees Number as Hyperparameter (BTHRTM)*

Something that can be done to try to improve Bagged Trees model predict capacity is to find best tree number with a cross-validation algorithm. Figure 4.15 shows results of this algorithm with **20** as optimal trees number.

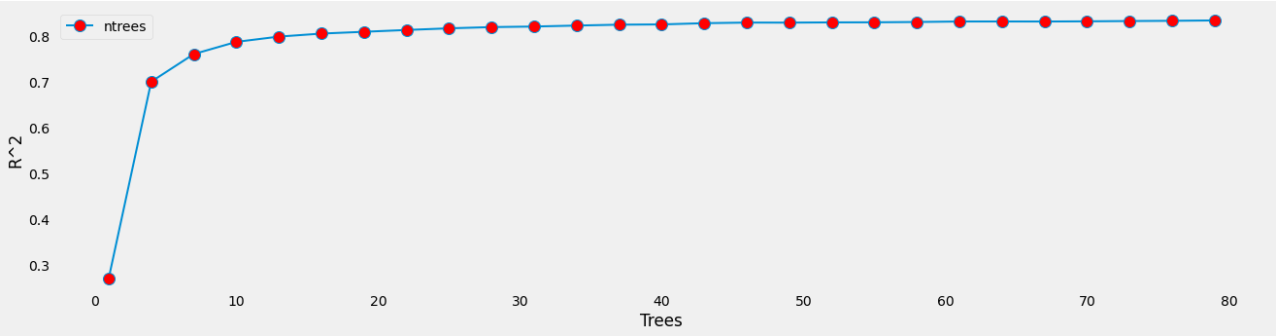

Figure 4.15: R^2 score VS. Trees number

Results of Regression Trees Models like this one will be presented on section 5.1.3.

4.5.5 *Random Forest Trees (RFRRTM)*

Random forest model usually performs well when previous models are not the best to predict our target variable. The number of base trees selected is **100**. The function to measure the quality of a split is the mean squared error. Minimum number of samples required to split an internal node and minimum number of samples required to be at a leaf node are defaults **2** and **1** respectively. The number of features

to consider when looking for the best split is **2**. Out-of-bag samples to estimate the generalization error will not be used. Results of Regression Trees Models like this one will be presented on section 5.1.3.

4.5.6 *Random Forest Trees with Best Trees Number as Hyperparameter (RTHRTM)*

To improve Random Forest Trees model performance is to find best tree number with a cross-validation algorithm. Figure 4.16 shows results of this algorithm with **20** as optimal trees number. Results of Regression Trees Models like this one will be presented on section 5.1.3.

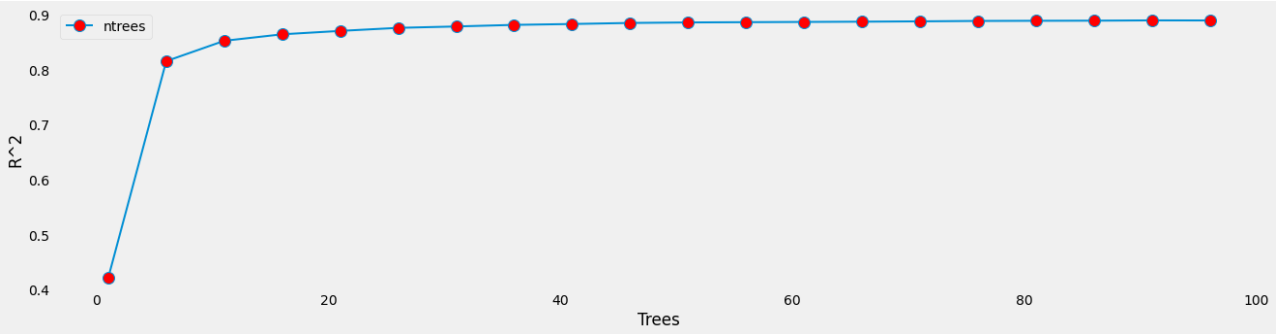

Figure 4.16: R^2 score VS. Trees number

5 Results and Discussion

Contents

5.1 Results **75**
5.1.1 Multiple Linear Regression Models Comparative 76
5.1.2 Support Vector Machine Regression Models Comparative 77
5.1.3 Regression Trees Models Comparative 78
5.2 Discussion **79**

5.1 Results

We conducted three schemes of regression models to predict our target variable **Moisture**. The first scheme uses multiple linear regression models with "raw" and standardized data. The second scheme uses SVM-based regression and the third scheme is Regression Trees models. The models of each scheme will be compared between them, selecting the one with the best performance, to subsequently compare them.

To compare models metrics used are coefficient of determination R^2, that can be interpreted as the proportion of the information in the data that is explained by the model. The root mean squared error (**RMSE**), usually interpreted as either how far (on average) the residuals are from zero or as the average distance between the observed values and the model predictions in the same nits as the original data, formally as 5.1:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\{y_n - t_n\}^2} \tag{5.1}$$

.

where t_n is the outcome and y_n is the model prediction of that sample's outcome.

Another metric used is the Mean Absolute Percentage Error (**MAPE**) that is the mean of all absolute percentage errors between the predicted

and actual values. It is a popular metric to use as it returns the error as a percentage, making it both easy for end users to understand and simple to compare model accuracy across use cases and datasets. Formally as equation 5.2:

$$MAPE = \frac{1}{N}\sum_{n=1}^{N}\frac{|y_n - t_n|}{t_n} \tag{5.2}$$

R^2 and **RMSE** will be evaluate on both train and test subsets in order to determine if model presents overfitting. Finally, **accuracy** metric is defined (as is understood in ALMEX context) as equation 5.3:

$$Accuracy = 100 - MAPE \tag{5.3}$$

Accuracy will be used on test subset to have a metric easier to explain to ALMEX operation team.

5.1.1 *Multiple Linear Regression Models Comparative*

The results of Multiple Linear Regression Models are shown in Table 5.1. All models shows consistent performance between train and test subsets discarding overfitting. In this case, **SRSLRM** is selected as the one that performs better, having the least **MAPE** and the highest R^2.

Table 5.1: Results of Multiple linear regression models

Model	Description	Tr RMSE	Ts RMSE	Tr R^2	Ts R^2	MAPE	Acc.
AFRLRM	All Features with Raw Data Linear Regression Mode	3.6846	3.9606	0.2128	0.1028	16.67	83.33
AFSLRM	All Features with Standardized Data Linear Regression Model	3.6846	3.9606	0.2128	0.1028	16.67	83.33
BESLRM	Backward Elimination with Standardized Data Linear Regression Model	3.6895	3.932	0.2117	0.1093	16.65	83.35
SRSLRM	Stepwise Regression with Standardized Data Linear Regression Model	3.7062	3.8786	0.2081	0.1214	16.56	83.44
OFSLRM	Operational Features with Standardized Data Linear Regression Model	4.3169	4.421	0.0776	-0.0024	17.78	82.22
PCALRM	Principal Component Linear Regression Model	3.7608	4.0023	0.1965	0.0933	16.87	83.13
PLSLRM	Partial Least Squares Regression Mode	3.6847	3.965	0.2127	0.1018	19.75	80.25

In addition, to understand how each model would perform on production stage, a trend plot for each multiple linear regression is

obtained overlapping predictions on the target variable. Production department main concern about predictions is that their behavior is the same as targets. Figure 5.1 shows that our selected model behavior is consistent (when the target increase, the prediction increase and viceversa).

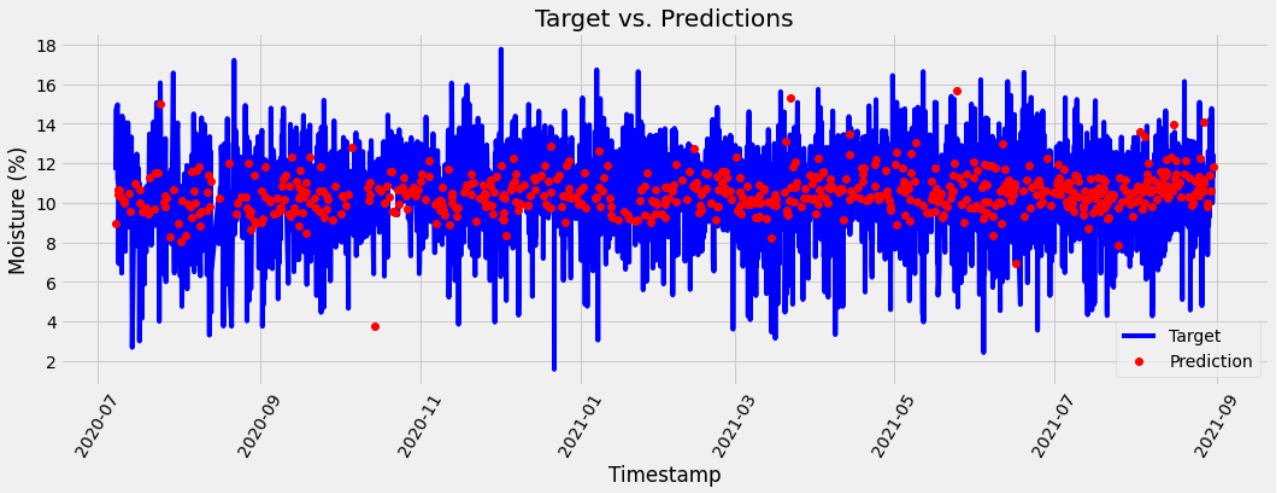

Figure 5.1: SRSLRM trend plot.

5.1.2 *Support Vector Machine Regression Models Comparative*

The results of Support Vector Machine Regression Models are shown in Table 5.2. All models shows consistent performance between train and test subsets discarding overfitting. In this case, **FRRSVM** is selected as the one that performs better, having the least **MAPE** and the highest R^2.

Model	Description	Tr RMSE	Ts RMSE	Tr R^2	Ts R^2	MAPE	Acc.
AFLSVM	All Features SVM Regression Model with Linear kernel	3.719	3.9121	0.2063	0.1138	16.67	83.33
AFRSVM	All Features SVM Regression Model with RBF kernel	3.0438	3.7078	0.3497	0.1601	16.38	83.62
FRLSVM	Feature Reduction SVM Regression Model with Linear kernel	3.7349	3.8635	0.2020	0.1248	16.75	83.25
FRRSVM	Feature Reduction SVM Regression Model with RBF kernel	3.0959	3.6976	0.3385	0.1624	16.28	83.72

Table 5.2: Results of Support Vector Machine regression models

Figure 5.2 shows that our selected model behavior is consistent on

selected model.

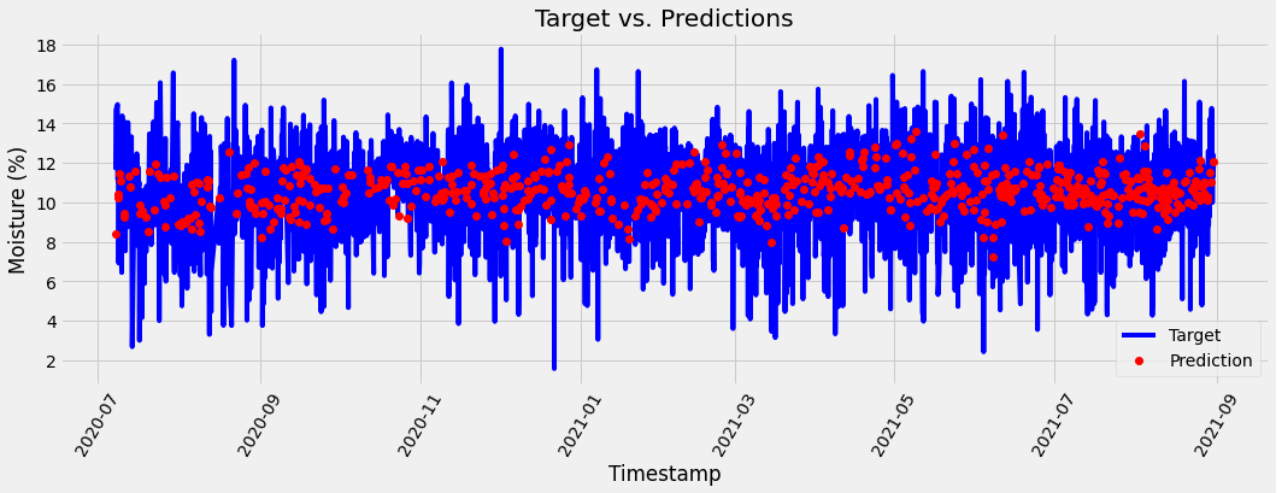

Figure 5.2: FRRSVM trend plot.

5.1.3 *Regression Trees Models Comparative*

The results of Regression Trees Models are shown in Table 5.3. Unlike other regression models, decision tree does not use regularization to fight against overfitting. Because there are few constraints placed on the decision tree algorithm's ability to learn new patterns, they are especially susceptible to overfitting[1] as we can see on our not pruned tree, so both Decision Trees models are discarded and decision is beetween bootstrapping models (bagged trees) and random forest. In this case, **BGGRTM** is selected as the one that performs better, having the least **MAPE** and the highest R^2.

[1] Tao Wang, Zhenxing Qin, Zhi Jin, and Shichao Zhang. Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning. *Journal of Systems and Software*, 83 (7):1137–1147, 2010. ISSN 0164-1212. DOI: https://doi.org/10.1016/j.jss.2010.01.002. SPLC 2008

Table 5.3: Results of Regression Trees models

Model	Description	Tr RMSE	Ts RMSE	Tr R^2	Ts R^2	MAPE	Acc.
DNDRTM	Decision Tree without Max Depth Defined	0.000	7.2011	1.0000	-0.6313	21.65	78.35
DDHRTM	Decision Tree with Best Depth as Hyperparameter	0.0491	6.9671	0.9895	-0.5783	21.09	78.91
BGGRTM	Bagged Trees	0.7962	3.6108	0.8299	0.1820	15.9	84.10
BTHRTM	Bagged Trees with Best Trees Number as Hyperparameter	0.8843	3.7267	0.8111	0.1558	16.1	83.90
RFRRTM	Random Forest Trees	0.5297	3.6925	0.8868	0.1635	16.2	83.80
RTHRTM	Random Forest Trees with Best Trees Number as Hyperparameter	0.6414	3.7916	0.8630	0.1411	16.4	83.60

Figure 5.3 shows that our selected model behavior is consistent on

target variable trend.

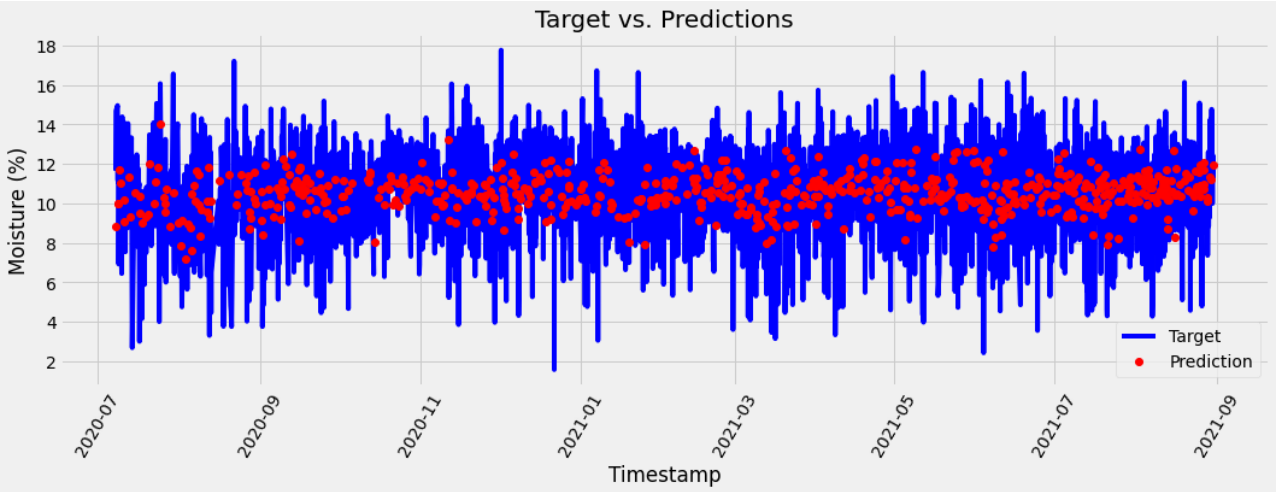

Figure 5.3: BGGRTM trend plot.

5.2 Discussion

The results of selected models of each scheme are shown in Table 5.4. The highest R^2 with the lowest **MAPE** is **BGGRTM**. Computational time near 1.4s is not a factor to choose another model, so this solution will be the one that will be presented to ALMEX's operation department. Despite prediction metrics are considered low and underfitted, **MAPE** is accepted because, for ALMEX, just **the prediction of target behavior can be used to act and try to control it**. This can be done through an alarm system that informs when actions taken predict if **moisture** decrease or increase as wished.

Model	Description	Tr RMSE	Ts RMSE	Tr R^2	Ts R^2	MAPE	Acc.
SRSLRM	Stepwise Regression with Standardized Data Linear Regression Model	3.7062	3.8786	0.2081	0.1214	16.56	83.44
FRRSVM	Feature Reduction SVM Regression Model with RBF kernel	3.0959	3.6976	0.3385	0.1624	16.28	83.72
BGGRTM	Bagged Trees	0.7962	3.6108	0.8299	0.1820	15.9	84.10

Table 5.4: Results of selected models of each scheme.

To complement, further work is needed to show a real-time solution in order operation team could take decisions about target variable **moisture** behaviour. Figure 5.4 displays the complete solution proposal

block diagram that will be presented to operation department in order to visualize the result of the selected model.

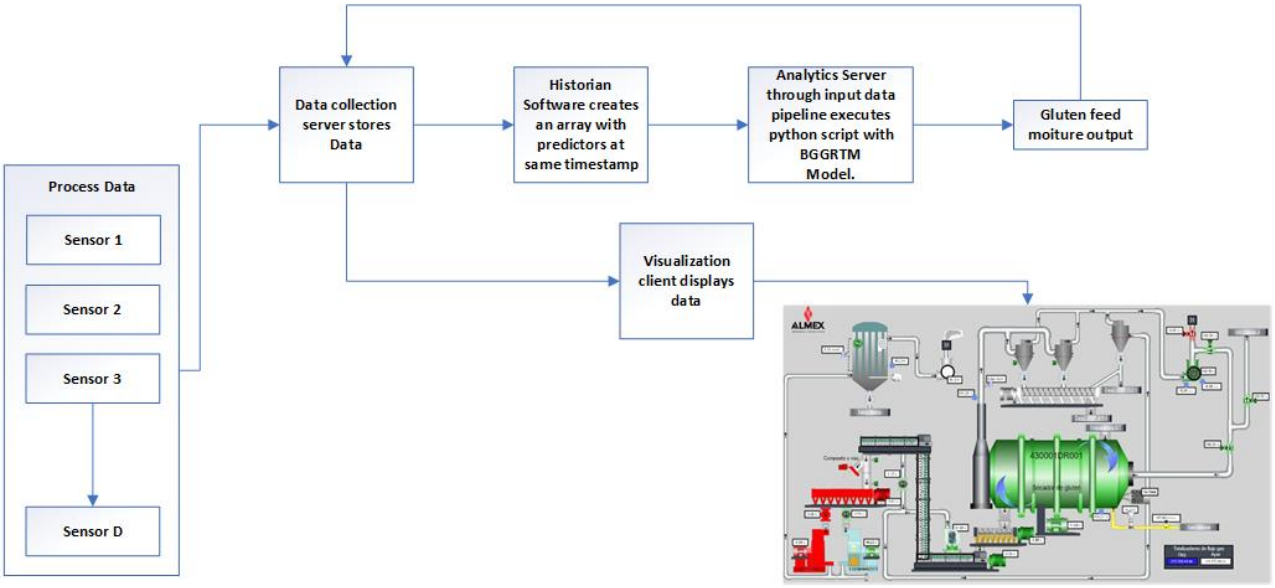

Figure 5.4: Block diagram of moisture visualization solution.

6 Conclusions

Contents

6.1 Conclusions 81
6.2 Future Work 81

6.1 Conclusions

This study presents Moisture of Corn Gluten Feed Drying Process regressor based on operational variables as a real-time and continuous monitoring features. Several machine learning algorithms were explored to predict with a database created with information proportioned by quality and production departments of ALMEX. However, analyzing Dataset used to train our models give us clues of a less than adequate operation in data entry. In some instances, entered **moisture** value does not make any operational sense, creating the idea that nor data is entered with correct sampling timestamp nor bad results are registered. Also, auditing timestamps on different operational shifts create patterns that indicate that **each QA analyst have their own data entry method and is not standardized**. As a result, a bagged trees-based regressor with low prediction score $\mathbf{R^2 = 0.1820}$ but an acceptable $\mathbf{MAPE = 15.9\%}$ is selected to predict target variable even with data registration faults. ALMEX agreed with this when notice that behavior plots confirms prediction following target on test subset and **accuracy is greater than 80%**. Thus, it can be concluded that this model is suitable to predict gluten feed **moisture** behaviour and, in order to improve model prediction score in the short term, future work, such as data entry operation standardization, is needed.

6.2 Future Work

Future efforts could be focused on improving the performance of the prediction models, by, for example, changing model architecture or

hyperparameter tuning. Analyzing Support Vector Machine results, our target variable does not seems to have a linear behavior, thus, efforts should be focused on algorithms that performs better with non-linear data. Looking section 2.2, the use of *neural network* algorithms to estimate *virtual sensors* shows that our prediction results could improve with this approach. Moreover, time series behaviour of **moisture** variable indicates that ARIMAX algorithms (Time Series Regression with exogenous variables) also would be suitable approach to improve our prediction performance. In addition to improve mentioned Machine Learning models solutions, and knowing that the ultimate goal is to control our studied variable, modern control theory algorithms and state observers approach could be used as they are similar to virtual sensors (not having a state feedback and have to estimate it). Finally, ALMEX requested a graphic representation for prediction model that will implemented outside this work scope, with the ML model approved by them.

Bibliography

Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010. ISSN 1939-0068. DOI: 10.1002/wics.101. URL http://dx.doi.org/10.1002/wics.101.

Douglas Adams. The hitchhiker's guide to the galaxy. 1995.

Suad A. Alasadi and Wesam S. Bhaya. Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12:4102–4107, sep 2017. DOI: 10.3923/jeasci.2017.4102.4107.

Serafín Alonso, Antonio Morán, Daniel Pérez, Perfecto Reguera, Ignacio Díaz, and Manuel Domínguez. Virtual sensor based on a deep learning approach for estimating efficiency in chillers. In *Communications in Computer and Information Science*, volume 1000, pages 307–319. Springer Verlag, 2019. ISBN 9783030202569. DOI: 10.1007/978-3-030-20257-6_26.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, NY, first edition, May 2006. ISBN 0-387-31073-8.

Paul Harwood Blanchard. *Technology of Corn Wet Milling and Associated Processes Volume 4 of Industrial chemistry library*. Elsevier, first edition, May 1992. ISBN 0-4448825-5-3.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. DOI: 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A%3A1010933404324.

Andrius Buteikis. Practical econometrics and data science. http://web.vu.lt/mif/a.buteikis/wp-content/uploads/PE_Book/3-2-OLS.html, October 2020.

Almidones Mexicanos S.A. de C.V. Comparison study beetween brimrose and perten online nir. Unpublished, dec 2016. Internal report (Almidones Mexicanos S.A. de C.V.).

Nikolay Dimitrov and Tuhfe Göçmen. Virtual sensors for wind turbines with machine learning-based time series models. *Wind Energy*, 25(9):1626–1645, 2022. DOI: https://doi.org/10.1002/we.2762. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/we.2762.

Mohamed Djerioui, Mohamed Bouamar, Mohamed Ladjal, and Azzedine Zerguine. Chlorine soft sensor based on extreme learning machine for water quality monitoring. *Arabian Journal for Science and Engineering*, 44, 2019. ISSN 2191-4281. DOI: 10.1007/s13369-018-3253-8. URL https://doi.org/10.1007/s13369-018-3253-8.

Norman R. Draper and Harry Smith. *Applied Regression Analysis*. Wiley-Interscience, third edition, 1998. ISBN 0-471-17082-8.

IBM Cloud Education. Machine learning. https://www.ibm.com/cloud/learn/machine-learning, July 2020.

Jacopo Foschi, Andrea Turolla, and Manuela Antonelli. Soft sensor predictor of e. coli concentration based on conventional monitoring parameters for wastewater disinfection control. *Water Research*, 191:116806, 2021. ISSN 0043-1354. DOI: https://doi.org/10.1016/j.watres.2021.116806. URL https://www.sciencedirect.com/science/article/pii/S004313542100004X.

Christie J. Geankoplis, A. Allen Hersel, and Daniel H. Lepek. *Transport processes & separation process principles*. Pearson, fifth edition, Jan 2018. ISBN 978-0-13-418102-8.

Paul Geladi and Bruce R. Kowalski. Partial least-squares regression: a tutorial. *Anal. Chim. Acta*, 185(C):1–17, 1986. DOI: 10.1016/0003-2670(86)80028-9.

Eushay Bin Ilyas, Marten Fischer, Thorben Iggena, and Ralf Tönjes. Virtual sensor creation to replace faulty sensors using automated machine learning techniques. In *2020 Global Internet of Things Summit (GIoTS)*, pages 1–6, 2020. DOI: 10.1109/GIOTS49054.2020.9119681.

Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, first edition, 2013. ISBN 978-1-4614-6849-3.

Dominik Martin, Niklas Kühl, and Gerhard Satzger. Virtual sensors. *Business & Information Systems Engineering*, 63(3):315–323, 2021. DOI: 10.1007/s12599-021-00689-w.

Benjamin Maschler, Sören Ganssloser, Andreas Hablizel, and Michael Weyrich. Deep learning based soft sensors for industrial machinery. *Procedia CIRP*, 99:662–667, 2021. ISSN 2212-8271. DOI: https://doi.org/10.1016/j.procir.2021.03.115. 14th CIRP Conference

on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020.

Daniele Masti, Daniele Bernardini, and Alberto Bemporad. A machine-learning approach to synthesize virtual sensors for parameter-varying systems. *European Journal of Control*, 61:40–49, 2021. ISSN 0947-3580. DOI: https://doi.org/10.1016/j.ejcon.2021.06.005. URL https://www.sciencedirect.com/science/article/pii/S0947358021000637.

William Mendenhall and Terry Sincich. *A second course in statistics : regression analysis*. Prentice Hall, seventh edition, 2012. ISBN 978-0-321-69169-9.

P.F. Muir. A virtual sensor approach to robot kinematic identification: theory and experimental implementation. In *1990 IEEE International Conference on Systems Engineering*, pages 440–445, 1990. DOI: 10.1109/ICSYSE.1990.203189.

UC Berkeley School of Information. What is machine learning (ml)? https://ischoolonline.berkeley.edu/blog/what-is-machine-learning, June 2020.

Arunima Sambhuta Pattanayak, Bhawani Shankar Pattnaik, Siba K. Udgata, and Ajit Kumar Panda. Development of chemical oxygen on demand (cod) soft sensor using edge intelligence. *IEEE Sensors Journal*, 20(24):14892–14902, 2020. DOI: 10.1109/JSEN.2020.3010134.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 01 2012.

Ivan Pisa, Ignacio Santín, Jose Lopez Vicario, Antoni Morell, and Ramon Vilanova. Ann-based soft sensor to predict effluent violations in wastewater treatment plants. *Sensors*, 19(6), 2019. ISSN 1424-8220. DOI: 10.3390/s19061280. URL https://www.mdpi.com/1424-8220/19/6/1280.

Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. ISSN 00063444. URL http://www.jstor.org/stable/2335739.

Kurt Schmidheiny. The multiple linear regression model. https://www.schmidheiny.name/teaching/ols.pdf, October 2022.

Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. *Proceedings of the 9th Python in Science Conference*, 2010, 01 2010.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965. ISSN 00063444. URL http://www.jstor.org/stable/2333709.

M. Stone and R. J. Brooks. Continuum regression: Cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(2):237–258, 1990. DOI: https://doi.org/10.1111/j.2517-6161.1990.tb01786.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1990.tb01786.x.

Agnes Tegen, Paul Davidsson, Radu-Casian Mihailescu, and Jan A. Persson. Collaborative sensing with interactive learning using dynamic intelligent virtual sensors. *Sensors*, 19(3), 2019. ISSN 1424-8220. DOI: 10.3390/s19030477. URL https://www.mdpi.com/1424-8220/19/3/477.

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

Tao Wang, Zhenxing Qin, Zhi Jin, and Shichao Zhang. Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning. *Journal of Systems and Software*, 83(7):1137–1147, 2010. ISSN 0164-1212. DOI: https://doi.org/10.1016/j.jss.2010.01.002. SPLC 2008.

S. Wold, H. Martens, and H. Wold. The multivariate calibration problem in chemistry solved by the pls method. In Bo Kågström and Axel Ruhe, editors, *Matrix Pencils*, pages 286–293, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg. ISBN 978-3-540-39447-1.

Jan Zenisek, Holger Gröning, Norbert Wild, Aziz Huskic, and Michael Affenzeller. Machine learning based data stream merging in additive manufacturing. *Procedia Computer Science*, 200:1422–1431, 2022. ISSN 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.01.343. 3rd International Conference on Industry 4.0 and Smart Manufacturing.

Index

Artificial Intelligence, 31

Backward Elimination, 36, 38, 66, 76
Bagged Tree, 48, 72, 78, 81

Corn Gluten Feed, 27, 81

Data Cleansing, 33, 54
Data Mining, 31
Data Preprocessing, 32, 53
Data Profiling, 33, 53
Data Reduction, 34, 55
Data Transformation, 34, 61
Data Validation, 62
Dewatering, 31
Drying, 28, 52

Homoscedasticity, 44, 64

Linear Regression, 21, 41, 42, 63, 75, 76

Machine Learning, 31, 32, 53, 81
MAPE, 20, 75, 81
Missing Values, 32, 33, 53, 54
Moisture, 28, 36, 52, 79

Ordinary Least Squares, 42, 63, 64

Partial Least Squares, 40, 70
Predictors, 33, 53
Principal Component Analysis, 38, 68

Random Forest Tree, 48, 73
Regression Tree, 46
RMSE, 76

Stepwise Regression, 36, 68, 79
Support Vector Machine, 25, 70, 77
Support Vector Machines, 44
Support Vectors, 46, 70

Target, 31, 35, 41, 52, 79

Virtual Sensors, 20, 23, 24

Wet Milling, 27, 28