



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA QUÍMICA  
ENG07053 - TRABALHO DE DIPLOMAÇÃO EM ENGENHARIA  
QUÍMICA



# Desenvolvimento de um Analisador Virtual para uma Indústria de Borracha Sintética

*Autor: Lucas Maciel*

*Orientador: Prof. Dr. Jorge Otávio Trierweiler*

*Coorientador: Me. Rafael Henrique Martello*

Porto Alegre, novembro de 2021



# Desenvolvimento de um Analisador Virtual para uma Indústria de Borracha Sintética

*Trabalho de Conclusão de Curso apresentado à COMGRAD/ENQ da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Bacharel em Engenharia Química*

Orientador: *Prof. Dr. Jorge Otávio Trierweiler*

Coorientador: Me. Rafael Henrique Martello

Banca Examinadora:

*Prof. Dr. Marcelo Farenzena, Universidade Federal do Rio Grande do Sul*

*Dr. Lucas Ranzan, Universidade Federal do Rio Grande do Sul*

Porto Alegre

2021

## AGRADECIMENTOS

Gostaria de agradecer primeiramente à Deus por ter iluminado a minha mente e me dado forças para seguir em frente mesmo nos momentos de dificuldade. Agradeço aos meus pais, Paulo e Luciane, e as minhas irmãs Bruna e Cacau por serem a minha base e estarem presentes em todos os momentos da minha vida. Também é parte essencial desta jornada o meu amor, Julia, que me acompanhou e apoiou em todos os momentos desde o início do curso de Engenharia, assim como a sua família: Marcos, Mariel e Luiza. Não posso deixar de agradecer também ao Professor Jorge que me orientou durante a realização deste trabalho e não mediu esforços para me ajudar, independentemente dos horários que as dúvidas surgissem. Sou muito grato também ao meu coorientador Rafael que forneceu ajuda valiosa durante a elaboração deste trabalho e me auxiliou nos meus primeiros passos em Python e com a Ciência de Dados. Agradeço também aos demais professores da UFRGS que fizeram parte da minha graduação e desempenharam com maestria a arte do ensinar. Fica também o agradecimento aos meus avós Brasil, Beatriz, Ivete e Osmar por todo o amor dispendido ao longo da minha criação. Aos amigos Eduardo, Emille, Guilherme, Nicole, Tiago e Tamizi pelos momentos felizes que ajudaram e muito ao longo da jornada. Enfim, a todas as pessoas que fizeram parte desta caminhada e que mesmo não citados neste agradecimento fazem parte do meu pensamento constantemente. Amo a todos vocês!

*“In the Middle of difficulty, lies Opportunity”.*

(Albert Einstein)

## RESUMO

Este trabalho visa o desenvolvimento de um analisador virtual para inferência da viscosidade Mooney durante o processo produtivo de borracha de butadieno e estireno (SBR), permitindo desta forma o uso de variáveis secundárias do processo coletadas a partir dos instrumentos existentes para determinação contínua desta variável que define a especificação ou não do produto. Primeiramente, foi realizada uma inspeção inicial nos dados para remoção de dados espúrios e, em seguida, foi empregada a Análise de Componentes Principais (PCA) para determinação da dimensionalidade do sistema, obtendo um valor aproximado do número de variáveis necessárias para compor o modelo de predição. Para a seleção das variáveis relacionadas com a viscosidade Mooney foi utilizado a regressão de Lasso, implementada em conjunto com o pacote Lars, que utiliza um critério de penalização para o coeficiente de regressão das variáveis de modo a manter diferente de zero somente as variáveis verdadeiramente relacionadas com a propriedade de interesse. Apesar de existirem técnicas mais sofisticadas para modelos de inferência, foi adotada a regressão linear multivariável (MLR) aplicada a uma tabela de dados estacionários onde as variáveis foram deslocadas de forma a compensar a dinâmica de planta. O deslocamento foi baseado no valor que levasse à máxima correlação com a variável de interesse. Para a estratégia de adaptação do modelo foi repetida a metodologia utilizada na obtenção do modelo para a viscosidade Mooney para obter um modelo simplificado de até 3 variáveis para inferência do erro ou resíduo gerado pelo modelo principal. Após somadas as predições dos modelos da viscosidade e dos resíduos, foi adotada uma estratégia de correção do bias de modo a somar o erro obtido entre as análises de laboratório e a predição do modelo neste mesmo instante. Depois de aplicado o algoritmo, em um conjunto de dados referentes a operação durante 5 meses da unidade de modo a calibrar os modelos e testar a qualidade das predições geradas, foi colocada em prova a capacidade preditiva do modelo sendo usado em um novo conjunto de dados de forma a simular a sua utilização na unidade industrial. Os resultados obtidos foram satisfatórios, de uma maneira geral, e, apresentaram um erro da ordem de 2,5% com relação ao valor médio de viscosidade Mooney do processo, apresentando-se o analisador virtual como uma útil ferramenta de controle do processo e fornece uma medida contínua da viscosidade Mooney do elastômero podendo ajudar na implementação de novas estratégias de operação, de modo a enviar de forma contínua o látex para a área de coagulação sem a necessidade de armazenamento em tanques de mistura, proporcionando economia devido ao menor consumo de vapor d'água no processo produtivo.

**Palavras-chave:** Borracha SBR, Analisador Virtual, Regressão Lasso, Regressão Linear

## ABSTRACT

This work aims to develop a virtual analyzer for Mooney viscosity inference during the production process of butadiene and styrene rubber (SBR), thus allowing the use of secondary process variables collected from existing instruments for continuous determination of this variable that defines the specification, or not, of the product. First, an initial inspection of the data was performed to remove spurious data, and then Principal Component Analysis (PCA) was used to determine the dimensionality of the system, obtaining an approximate value of the number of variables needed to compose the model. For the selection of variables related to Mooney viscosity, Lasso regression was used, implemented together with the Lars package, which uses a penalty criterion for the regression coefficient of unrelated variables to keep only the true variables different from zero relating to the property of interest. Although there are more sophisticated techniques for inference models, multivariable linear regression (MLR) was adopted, applied to a stationary data table where the variables were shifted to compensate the plant dynamics. The displacement was based on the value that led to the maximum correlation with the variable of interest. For the model adaptation strategy, the procedure used to develop the model of the main variable was repeated to obtain a simplified model of up to 3 variables for inference of the error or residual generated by the main model. After adding the predictions of the viscosity and residuals models, a bias correction strategy was adopted to add the error obtained between the laboratory analyzes and the prediction of the model at this time. After applying the algorithm, in a dataset referring to the unit's operation for 5 months to calibrate the models and test the quality of the generated predictions, the predictive capacity of the model being used in a new dataset was tested to simulate its use in the industrial unit. The results were generally satisfactory, and despite showing an error of the order of 2.5% in relation to the average Mooney viscosity value of the process, the virtual analyzer presents itself as a useful tool for controlling the process and provides a continuous measure of the elastomer's Mooney viscosity, which can help in the implementation of new operating strategies, in order to continuously send the latex to the coagulation area without the need for storage in mixing tanks, providing savings due to lower consumption of water steam in the production process.

**Keywords:** *SBR Rubber, Virtual analyzer, Lasso Regression, Linear Regression*

## LISTA DE FIGURAS

Figura 1: Estrutura básica de um analisador virtual .....	2
Figura 2: Representação gráfica do problema de MLR.....	9
Figura 3: Estudo do deslocamento de tempo ótimo .....	14
Figura 4: Desenho ilustrativo do processo de fabricação de borracha sintética.....	17
Figura 5: Fluxograma de Processo da unidade de polimerização .....	22
Figura 6: Fluxograma simplificado do sistema de refrigeração dos reatores .....	23
Figura 7: Variância explicada em função do número de principais componentes.....	26
Figura 8: Análise do impacto do fator de penalidade no valor de $R^2$ .....	26
Figura 9: Comparação entre os três modelos obtidos: MLR, <i>Gradient Boosting</i> e <i>XGBRegressor</i> .....	28
Figura 10: Melhor predição para a variável 42 .....	30
Figura 11: Ajuste obtido para o modelo dos resíduos.....	31
Figura 12: Resultado da predição obtida através da estratégia do Caso B .....	32
Figura 13: Resultado de predição da viscosidade Mooney e comparação com resultados de laboratório .....	33
Figura A.1: Etapas da Polimerização em emulsão via radicais livres .....	40



## LISTA DE TABELAS

<b>Tabela 1:</b> Descrição dos equipamentos de processo. ....	19
<b>Tabela 2:</b> Descrição das variáveis de processo. ....	20
<b>Tabela 3:</b> Deslocamento de tempo ótimo para as variáveis do conjunto de dados com localização a montante do ponto de coleta .....	25
<b>Tabela 4:</b> Valores de $R^2$ e $n^\circ$ de variáveis com coeficiente de regressão diferente de zero em função do parâmetro de penalidade alpha .....	27
<b>Tabela 5:</b> Resultados de predição para os três modelos testados.....	28
<b>Tabela 6:</b> Comparação de resultados para inclusão de variáveis .....	29
<b>Tabela 7:</b> Resultados das estratégias de adaptação implementadas .....	32
<b>Tabela 8:</b> Comparativo dos resultados para diferentes frequências de análises .....	34
<b>Tabela A.1:</b> Condições de Processo e componentes utilizados na polimerização a frio de SBR .....	40

## LISTA DE ABREVIATURAS E SIGLAS

PCA – *Principal Component Analysis*

PLS – *Partial Least Squares*

SSE – *Sum of Squared Errors*

MAE – *Mean Absolute Error*

MSE – *Mean Squared Error*

RMSE – *Root Mean Squared Error*

AIC – *Akaike Information Criterion*

BIC – *Bayesian Information Criterion*

BE – Busca Exaustiva

VS – ACO – *Variable Selection Ant Colony Optimization*

SVD – *Singular Value Decomposition*

LASSO – *Least Absolute Shrinkage and Selection Operator*

FS - *Forward Selection*

MLR – *Multiple Linear Regression*

PCR – *Principal Component Regression*

CSTR – *Continuous Stirred Tank Reactor*

LARS – *Least Angle Regression*

SBR – *Styrene-Butadiene Rubber*

VM – Viscosidade Mooney

## SUMÁRIO

1	Introdução	1
1.1	Analisadores Virtuais	2
1.2	Técnicas de Modelagem	2
1.3	Desenvolvimento de Analisadores Virtuais	3
2	Revisão Bibliográfica e Fundamentação Teórica	4
2.1	Seleção de Variáveis	4
2.1.1	dMétricas para avaliação de Modelos	4
2.1.2	PCA – Principal Component Analysis	5
2.1.3	Regressão Lasso	6
2.2	Escolha do Modelo de Inferência	7
2.2.1	Regressão Linear Multivariável	8
2.2.2	Gradient Boosting Regressor	9
2.3	Estratégia de Adaptação	11
3	Metodologia Proposta	12
3.1	Seleção de Variáveis	12
3.1.1	Primeira Inspeção dos Dados	13
3.1.2	Deslocamento temporal de variáveis	13
3.1.3	Determinação da dimensionalidade do Problema e Seleção das Variáveis Secundárias	14
3.2	Elaboração do Modelo de Inferência	14
3.3	Estratégia de Adaptação	15
4	Estudo de Caso	16
4.1	Descrição da Unidade de Polimerização de borracha SBR	16
4.2	Variáveis medidas em linha	19
4.3	Variável medida em laboratório: Viscosidade Mooney (VM)	23
5	Resultados e Discussões	25
5.1	Variáveis Escolhidas	25
5.2	Modelos Finais Obtidos	28
5.3	Estratégias de adaptação – modelo do resíduo	30
5.4	Validação do modelo	33
5.5	Comparação com a literatura existente	34
6	Conclusão e sugestões para trabalhos futuros	35
	REFERÊNCIAS	36
	APÊNDICE A	39
	A.1 Borracha de Estireno-Butadieno (SBR)	39

A.1.1 Polimerização em Emulsão a frio	39
A.1.2 Etapas da polimerização em emulsão a frio via radicais livres	40
A.1.3 Características controladas na reação de SBR	41
APÊNDICE B	42
B.1 Código de Implementação da Metodologia em Python	42

## 1 Introdução

O incremento da competitividade no cenário econômico atual exige estratégias mais sofisticadas de controle de processo a fim de otimizar a produção, reduzir a obtenção de produtos fora de especificação e permitir que as empresas ingressem no mercado com produtos a preços mais competitivos. Um dos principais limitantes para um controle rígido de determinado processo é a impossibilidade de se monitorar de maneira contínua a variável que determina a especificação do produto (FORTUNA; GRAZIANI; XIBILIA, 2005). Na maioria dos casos esta variável é obtida a partir de análises em laboratório espaçadas no tempo ou a partir de instrumentos instalados em linha que também exigem determinado intervalo de tempo entre a coleta da amostra e a conclusão da análise (cromatografia, por exemplo). Este é o caso da viscosidade Mooney, que é uma variável relacionada à distribuição de peso molecular de elastômeros e que é a principal variável controlada no processo produtivo de borracha SBR (*Styrene and Butadiene Rubber* ou em português, borracha de estireno e butadieno) estudado neste trabalho, pois está diretamente relacionada às características de processamento da borracha nas etapas seguintes de vulcanização nas indústrias de terceira geração.

Devido à grande disponibilidade de dados obtidos a partir dos sensores (instrumentos) de processo os analisadores virtuais surgem como uma excelente possibilidade como estratégia de monitoramento contínuo de processos industriais. Os analisadores virtuais são algoritmos desenvolvidos para estimar variáveis de processo a partir das medidas de e variáveis secundárias que sejam correlacionadas com a variável que se deseja monitorar (FORTUNA, 2007). Desta forma, o objetivo geral deste trabalho é o desenvolvimento de um analisador virtual para utilização no processo de produção de borracha SBR de modo a proporcionar um monitoramento contínuo da viscosidade Mooney obtido ao final da cadeia de reatores de polimerização. O analisador virtual permitirá a otimização do controle operacional da planta em questão (a planta será detalhada no capítulo de Estudo de Caso):

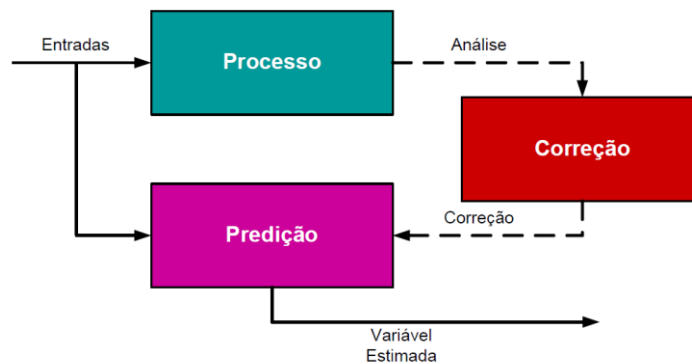
- Obter um analisador virtual de fácil implementação para que a partir dele os operadores de processo tenham uma ferramenta útil para controle do processo;
- A partir do monitoramento contínuo da viscosidade Mooney poderá ser reduzido o tempo de resposta a desvios no processo, permitindo ações antecipadas por parte dos operadores para que realizem as correções necessárias durante a operação da planta.
- Estudar a possibilidade de redução da frequência de análises de laboratório realizadas, com base nos resultados obtidos no conjunto de dados de validação do modelo comparando estratégias de correção do bias com amostras de diferentes intervalos de tempo.

A seguir será discutido brevemente o que são os analisadores virtuais e as principais técnicas de modelagem existentes.

## 1.1 Analisadores Virtuais

Analisadores virtuais ou sensores por software (*soft sensors*) podem ser definidos como algoritmos matemáticos capazes de inferir em tempo real variáveis de interesse não medidas, como, por exemplo, índices de fluidez para reatores de polimerização e composição de colunas de destilação, a partir de outras variáveis secundárias de processo medidas instantaneamente por instrumentos (WERLE, 2012).

Figura 1: Estrutura básica de um analisador virtual



Fonte: Extraído de (FACCHIN, 2005).

A Figura 1 mostra os componentes de um analisador virtual: o bloco Processo consiste nos dados de processo provenientes de instrumentos de medição disponíveis, o bloco Predição contém o modelo matemático que relaciona as variáveis de processo com a propriedade a ser estimada e o bloco Correção é composto por uma estratégia de ajuste do modelo a partir dos resultados de laboratório obtidos (FACCHIN, 2005).

Além da possibilidade de antecipação do resultado laboratorial, permitindo uma atuação mais ágil sobre o processo, o analisador virtual após implementado não depende grandes custos quando comparados as análises laboratoriais, que exigem mão-de-obra, insumos, infraestrutura etc. (MOHR, 2004). O desenvolvimento de um analisador virtual pode ser dividido em cinco etapas. Primeiramente é realizada uma inspeção nos dados para visualização da sua estrutura e identificação de dados problemáticos ou faltantes. Após, são selecionadas as variáveis para ajuste do modelo e são identificadas as áreas de operação em regime estacionário. Na terceira etapa é realizada uma filtragem dos dados, removendo-se *outliers* e realizando o escalonamento apropriado. Na quarta fase é realizado o ajuste do modelo utilizado para inferência e por último realiza-se a manutenção contínua e ajuste fino do *soft sensor* (MENEGOLLA, 2019).

## 1.2 Técnicas de Modelagem

O grande desafio no desenvolvimento de analisadores virtuais é a construção do modelo matemático que represente o processo em questão e para isso existem três diferentes abordagens: modelagem fenomenológica, modelagem por estatística multivariável e modelagem por inteligência artificial, tais como as redes neuronais. Este tipo de classificação não deve ser encarado de forma rígida pois as metodologias típicas de cada uma delas pode ser melhorada por técnicas pertencentes as demais abordagens (FORTUNA, L. et al., 2007). A

modelagem fenomenológica também conhecida como modelos caixa-branca exige o conhecimento profundo do processo em questão e das leis da física que o regem (WERLE, 2012). Como exemplos de técnicas de modelagem por estatística multivariável podemos citar os métodos de MLR (*Multiple Linear Regression*), PCA (*Principal Component Analysis*) e PLS (*Partial Least Squares*), sendo que frequentemente estas técnicas são utilizadas em conjunto com as Redes Neurais para estudos preliminares dos dados disponíveis (FACCHIN, 2005). Devido ao avanço no que diz respeito a instrumentação e automação dos processos industriais, levando a disponibilidade a numerosos bancos de dados de processo os modelos baseados em modelos de estatística multivariável ou em redes neurais, os chamados modelos caixa-preta, vem ganhando cada vez mais aplicação por exigirem somente a aquisição de um conjunto de dados de tamanho adequado para treinamento do modelo de modo a representar o processo (WERLE, 2012). Devido a maior praticidade na sua aplicação e a disponibilidade de uma extensa base de dados de processo, a abordagem caixa-preta será utilizada na elaboração do presente trabalho.

### 1.3 Desenvolvimento de Analisadores Virtuais

O desenvolvimento do texto deste trabalho será orientado pelos três pilares ou etapas principais do desenvolvimento de analisadores virtuais que consistem em:

- **Seleção de Variáveis:** esta etapa é importante para o desenvolvimento de analisadores virtuais, pois a escolha de variáveis que verdadeiramente se correlacionem com a propriedade a ser estimada está intrinsicamente relacionada com a qualidade do modelo desenvolvido. Após a coleta de todos os dados de processo disponíveis e da realização de uma primeira filtragem para remoção de dados faltantes ou *outliers* (oriundos de falhas de sensores, por exemplo), são empregadas metodologias disponíveis na literatura para que então sejam selecionadas as variáveis que irão compor o modelo. Podem ainda ser desenvolvidas novas variáveis a partir de operações matemáticas realizadas nas variáveis disponíveis ou da combinação entre elas, baseadas em conhecimento de engenharia por exemplo. Esta ação visa fornecer ao modelo comportamentos esperados pelo conhecimento prático ou teórico do processo, mas que não foram possíveis de serem detectados pelas variáveis de processo diretamente.
- **Elaboração do modelo de inferência:** após a seleção das variáveis a serem utilizadas, são então testados um ou mais modelos disponíveis na literatura para verificar a sua capacidade de predição dos dados de processo. Para isso, os dados de processo são divididos em dois grupos: o primeiro grupo denominado de treinamento ou calibração é utilizado para ajustar os parâmetros do modelo, sendo fornecido tanto as variáveis secundárias como os resultados os quais se deseja estimar. Já o segundo conjunto de dados, denominado de grupo de teste é utilizado para testar a qualidade de predição do modelo, calculando, a partir das variáveis secundárias e dos parâmetros do modelo, a propriedade a ser estimada. Então, a qualidade do modelo é avaliada a partir de diversas métricas disponíveis, através da avaliação do erro entre os valores alvo e os valores preditos.
- **Implementação de uma estratégia de adaptação do modelo:** devido à natureza dos processos industriais, como mudanças nas matérias-primas, por exemplo, ou

desgaste nos equipamentos e ou instrumentos, a tendência natural é a contínua perda de qualidade do modelo gerado, sendo aconselhada a implementação de uma sistemática de correção dos parâmetros de modelo ou dos resultados de predição, a partir de valores atuais da variável estimada (resultados das análises de laboratório, por exemplo). Na seção de revisão bibliográfica e fundamentação teórica serão abordadas as técnicas disponíveis na literatura e as técnicas utilizadas na elaboração do presente trabalho.

## 2 Revisão Bibliográfica e Fundamentação Teórica

Nesta seção serão abordados os conceitos teóricos envolvidos na elaboração deste trabalho. Como já citado na seção 1.2, o modelo desenvolvido no presente trabalho focará na metodologia caixa-preta devido à complexidade do processo e à disponibilidade de uma extensa base de dados. A seguir será realizada uma revisão da literatura disponível, relativa aos três pilares fundamentais do desenvolvimento de analisadores virtuais.

### 2.1 Seleção de Variáveis

A seleção de variáveis é de fundamental importância na construção de um modelo, sendo responsável, ou não, pelo sucesso da modelagem. Todo o modelo construído para representar determinado processo carregará erros, sendo impossível obter um modelo perfeito. Desta forma, o foco do desenvolvimento de um modelo deve ser a obtenção de um modelo que forneça uma aproximação coerente da realidade e que, principalmente, seja útil para auxiliar o controle do processo (BROWNE, 2000). Para alcançar este objetivo, o modelo deve satisfazer a dois principais requisitos: (i) ser claramente compreendido pelo seu desenvolvedor e (ii) se ajustar com um suficiente grau de precisão à realidade. Para que o segundo requisito seja atendido existem na literatura uma série de metodologias de seleção de variáveis e métricas para avaliação de modelos de inferência, porém para que o primeiro requisito seja correspondido deve ser realizada uma criteriosa análise das variáveis selecionadas para verificação se elas possuem sentido físico e prático (BROWNE, 2000).

#### 2.1.1 *dMétricas para avaliação de Modelos*

Dentre os diversos índices de ajuste ou métricas para avaliação de modelos de predição, serão abordados a seguir alguns dos mais populares e que foram utilizados na elaboração do presente trabalho. Apesar da existência de uma variada gama de métricas distintas, basicamente todas são baseadas na soma quadrática do erro, SSE, como parâmetro do desvio dos dados preditos pelo modelo e os dados reais (FACCHIN, 2005), definido pela seguinte equação:

$$SSE = \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.1)$$

Onde  $N$  = número de amostras,  $y_i$  = valor da amostra,  $\hat{y}_i$  = valor obtido pelo modelo,  $\bar{y}_i$  = média do conjunto de calibração.

O índice mais popular para medir a capacidade de ajuste de um modelo é o coeficiente de determinação, normalmente referenciado como  $R^2$ . Essa métrica utiliza uma relação entre o erro originado pelo modelo e a distância de cada ponto à média do conjunto de calibração, sendo este fator representado por:



$$S_{yy} = \sum_{i=1}^n (y_i - \bar{y}_i)^2 \quad (2.2)$$

Como a razão entre as expressões 2.1 e 2.2 é complementar com relação ao valor unitário, a expressão final para o  $R^2$  pode ser expressa por:

$$R^2 = 1 - \frac{SSE}{S_{yy}} \quad (2.3)$$

O uso do coeficiente de determinação deve ser realizado com cuidado, pois apresenta crescimento assintótico até o valor unitário a partir da inclusão de novas variáveis ao modelo (MONTGOMERY; PECK, 1982) e se utilizado para modelos lineares sem o termo de interceptação ou em modelos não-lineares os diferentes pacotes existentes para o cálculo deste termo poderão reproduzir valores distintos (KVALSETH, 1985).

Podem ser destacados ainda outros índices de ajuste, tais como o erro médio absoluto (MAE), o erro médio quadrático (MSE) e a sua respectiva raiz quadrada (RMSE) e o índice da variância explicada (*explained variance score*), cujas respectivas equações estão expostas a seguir:

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (2.4)$$

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (2.5)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (2.6)$$

$$\exp\_var = 1 - \frac{\sigma^2(y_i - \hat{y}_i)}{\sigma^2(y_i)} \quad (2.7)$$

Onde  $\sigma^2(y_i - \hat{y}_i)$  = variância ou quadrado do desvio padrão de  $(y_i - \hat{y}_i)$  e  $\sigma^2(y_i)$  = variância ou quadrado do desvio padrão de  $y_i$ .

Dentre os métodos existentes para a seleção de variáveis, destacam-se os métodos de Busca Exaustiva (BE), que testam todas as possíveis combinações entre as variáveis disponíveis e por isso garantem a escolha do melhor conjunto possível de variáveis para compor o modelo. Porém este método não é indicado para sistemas com elevado número de variáveis, por exigir uma elevada carga de processamento computacional (RANZAN *et al.*, 2015). Existem também diversos trabalhos focados em desenvolver métodos em que não seja necessário o teste de todas as combinações como, por exemplo, Facchin (2005) que comparou o uso de técnicas de algoritmos genéticos com BE sem grande sucesso e Ranzan *et al.* (2015) que comparam uma nova metodologia desenvolvida para a seleção de variáveis baseada em técnicas de otimização Colônia de Formigas (VS-ACO) e que demonstraram resultados compatíveis com os métodos de BE com um tempo computacional inferior para um conjunto de até 20 variáveis.

### 2.1.2 PCA – Principal Component Analysis

A análise de componentes principais (em inglês *Principal Component Analysis – PCA*) é um método para análise multivariável largamente utilizado visando à redução da

dimensionalidade de conjuntos de dados, na detecção de outliers, na classificação e agrupamento gráfico de dados, entre diversas outras funcionalidades (GELADI; LINDERHOLM, 2020). O principal objetivo deste método é extrair informação relevante do conjunto de dados e expressar esta informação como um novo conjunto de variáveis ortogonais entre si denominadas de componentes principais (ABDI; WILLIAMS, 2010). A decomposição da matriz de dados original em uma soma de matrizes de posto unitários e ortogonais entre si elimina a presença de dados altamente correlacionados, fator que é prejudicial no uso de regressão linear multivariável (FACCHIN, 2005). A abordagem matemática do cálculo dos principais componentes a partir de SVD (*Singular Value Decomposition*) pode ser encontrada em diversos trabalhos: (GELADI; LINDERHOLM, 2020), (ABDI; WILLIAMS, 2010) e (VAN DEN BERG, 2003).

Apesar do PCA possuir os mais variados usos associados com campos específicos da ciência, de forma geral os seus objetivos podem ser resumidos em 3 principais (VAN DEN BERG, 2003):

- Determinação de fatores ou “pseudo-variáveis” denominados de componentes principais, a partir de um conjunto de dados, realizando combinações lineares entre as variáveis originais;
- Os componentes principais são selecionados na ordem que os permita explicar o máximo de informação (variância) presente no conjunto de dados;
- As novas variáveis eliminam informações redundantes e filtram o ruído do conjunto de dados original.

O fato de os componentes principais explicarem a variância dos dados, eliminarem informações redundantes e filtrarem o ruído fornece uma noção da dimensionalidade do sistema (variáveis independentes), podendo esta técnica ser utilizada para fornecer uma noção do número de variáveis necessárias para elaboração de um modelo de inferência de determinado conjunto de dados (MENEGOLLA, 2019).

### 2.1.3 Regressão Lasso

A regressão Lasso, do inglês *Least Absolute Shrinkage and Selection Operator* ou operador de seleção e redução mínima absoluta foi proposto por Robert Tibshirani em 1996 e consiste em um método poderoso para regularização e seleção de variáveis (FONTI, 2017). O método emprega uma penalidade do tipo L1 nos coeficientes de regressão, escolhendo entre um conjunto de variáveis altamente correlacionadas apenas uma e zerando o coeficiente das demais, o que tende a produzir modelos esparsos e por este motivo é frequentemente utilizado como ferramenta para seleção de variáveis (LENG; LIN; WAHBA, 2006). O método Lasso impõe uma restrição na soma dos valores absolutos dos parâmetros do modelo, ou seja, esta soma necessita ser inferior a restrição imposta. Para isto, o método aplica um processo de redução (regularização) onde ocorre a penalização dos coeficientes de regressão das variáveis reduzindo alguns deles a zero. Ao longo do processo de seleção de variáveis, as variáveis cujo coeficientes permanecerem com valor diferente de zero após o processo de redução são selecionadas para comporem o modelo (FONTI, 2017).

Com relação a formulação matemática, a estimativa de lasso é definida pela solução do problema de otimização L1:

$$\min_{\beta} \left( \frac{\|Y - X\beta\|_2^2}{n} \right) \quad \text{sujeito a } \sum_{j=1}^k \|\beta_j\|_1 < t \quad (2.8)$$

Onde  $t$  é o limite superior para a soma dos coeficientes e o  $\beta$  são os valores dos coeficientes de regressão. De forma análoga, para o parâmetro de estimação podemos escrever:

$$\hat{\beta}(\lambda) = \operatorname{argmin}_{\beta} \left( \frac{\|Y - X\beta\|_2^2}{n} + \lambda \|\beta\|_1 \right) \quad (2.9)$$

Onde  $\|Y - X\beta\|_2^2 = \sum_{i=0}^n (Y_i - (X\beta)_i)^2$ ,  $\|\beta\|_1 = \sum_{j=1}^k |\beta_j|$  e  $\lambda \geq 0$ . O  $\lambda$  é o parâmetro que regula a intensidade da penalidade, ou seja, quanto maior o seu valor maior será o número de variáveis com coeficiente de regressão reduzido a zero. Vale ressaltar que o parâmetro  $\lambda$  e o limite superior para a soma dos coeficientes  $t$  são inversamente proporcionais: para valor de  $t$  tendendo ao infinito,  $\lambda$  tende a zero, tornando o problema um caso de mínimo quadrados comum. Isso vale para a situação oposta, para valores de  $t$  tendendo a zero o valor de  $\lambda$  tenderá a infinito e todos os coeficientes de regressão das variáveis serão reduzidos a zero (FONTI, 2017).

Com relação à seleção de variáveis, quando o problema de otimização (2.9) é minimizado, alguns dos coeficientes de regressão são reduzidos a zero, ou seja,  $\hat{\beta}_j(\lambda) = 0$ , para alguns valores de  $j$ . Com isso, as variáveis cujo coeficiente foi reduzido a zero são removidas do modelo e este fato torna o método de lasso uma poderosa ferramenta para a seleção de variáveis (FONTI, 2017).

Existem alguns pacotes de implementação do método lasso para o seu uso para seleção de modelos, sendo o Lars, do inglês *Least Angle Regression* ou regressão de ângulos mínimos o utilizado no presente trabalho. O método Lars é baseado no mecanismo tradicional de *Forward Selection (FS)*, onde é selecionada a variável com maior valor absoluto de correlação com a variável a ser estimada (FONTI, 2017). No caso da sua implementação em conjunto com o método LASSO este procedimento será repetido até que a restrição imposta seja alcançada, sendo as variáveis com coeficiente igual a zero descartadas.

## 2.2 Escolha do Modelo de Inferência

Existem três classes principais de modelagem a serem consideradas (CONZ, 2005):

- **Modelos fenomenológicos:** utilizam modelos teóricos baseados no profundo conhecimento de determinado processo para relacionar as variáveis secundárias com determinada variável a ser estimada, fato este que dificulta a sua utilização em processos industriais complexos onde existem muitas variáveis correlacionadas. São exemplos de modelos fenomenológicos os balanços de massa e energia, as leis da termodinâmica, equações de estado, cinéticas de reação, entre muitas outras.
- **Modelos empíricos:** trata-se de modelos que utilizam técnicas de análise estatística mono ou multivariável de modelos de regressão para estabelecer uma

relação entre os dados de entrada e saída. Devido a abundância de dados disponíveis e a facilidade de implementação esta será a classe de modelo empregada neste trabalho.

- **Modelos semiempíricos:** são modelos oriundos de simplificações realizadas em modelos fenomenológicos que geram parâmetros que necessitam de ajuste pelos métodos estatísticos para que possam ser utilizados na estimação de parâmetros.

Dentro dos modelos empíricos, os métodos estatísticos multivariáveis, baseados em projeção linear são largamente estudados e utilizados, principalmente quando os dados do processo apresentam elevada dimensionalidade e colinearidade (WERLE, 2012). Dentre os métodos baseados em projeções lineares podem ser destacados a Regressão Linear Múltipla (MLR), a Regressão de Principais Componentes (PCR) e os Quadrados Mínimos Parciais (PLS), pois são fáceis de implementar e apresentam resultados satisfatórios (WERLE, 2012).

Como os processos industriais tendem a possuir comportamentos não-lineares em suas variáveis, os modelos lineares podem não apresentar bons resultados em determinados casos. Para isso, foram desenvolvidos métodos mais robustos com capacidade de modelar estas não-linearidades, como o Filtro de Kalman Estendido (ROCHA, 2019), algumas técnicas de PLS não-lineares (LIU *et al.*, 2019) e as redes neurais (WERLE, 2012). Devido, principalmente, à facilidade de implementação, os modelos testados neste trabalho foram o MLR e dois modelos baseados em algoritmos de árvores de decisões que surgem como alternativas para tentar capturar o comportamento não-linear das variáveis que compõem o modelo.

### 2.2.1 Regressão Linear Multivariável

A regressão Linear Multivariável (MLR) pode ser definida como uma extensão da regressão linear simples para um conjunto de dados com mais de uma variável independente ( $x$ ). O problema de MLR pode ser definido a partir de um conjunto de variáveis independentes ou explicativas  $\mathbf{X}$ , composto por  $m$  colunas e  $n$  linhas e um vetor correspondente a variável dependente  $\mathbf{y}$ , composto por 1 coluna e  $n$  linhas deseja-se definir uma correlação linear entre elas, ou seja (GELADI; KOWALSKI, 1986):

$$y = b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 + \dots + b_m \cdot x_m + e \quad (2.10)$$

Ou de forma condensada:

$$y = \sum_{j=1}^m b_j \cdot x_j + e = \mathbf{x}^T \cdot \mathbf{b} + e \quad (2.11)$$

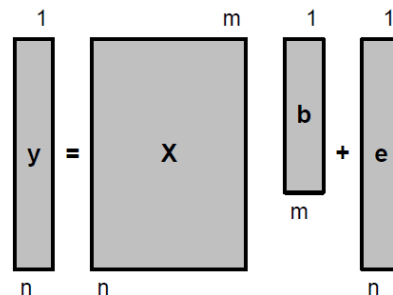
A equação (2.11) descreve as dependências multilineares para variáveis com tamanho de amostra  $n=1$ . Para amostras com  $n>1$  deve ser utilizada a notação vetorial (GELADI; KOWALSKI, 1986).

$$y = \mathbf{X} \cdot \mathbf{b} + e \quad (2.12)$$

Onde  $\mathbf{X}$  é uma matriz de variáveis secundárias (com  $n$  linhas e  $m$  colunas),  $\mathbf{b}$  é o vetor de coeficientes de cada variável secundária e a variável “ $e$ ” é o vetor dos resíduos. Segundo

(GELADI; KOWALSKI, 1986), uma representação gráfica do problema de MLR pode facilitar a compreensão dos leitores, como mostra a Figura 2:

Figura 2: Representação gráfica do problema de MLR



Fonte: (FACCHIN, 2005).

Dentro do problema de MLR podem ocorrer três situações (GELADI; KOWALSKI, 1986):

- $m > n$ : existe um maior número de variáveis do que amostras para cada variável. Neste caso caímos em uma situação que não é útil do ponto de vista prático, pois existe um infinito número de soluções para  $b$  que se ajustam a equação 2.12.
- $m = n$ : o número de variáveis e de amostras para cada variável são iguais. Neste caso existe uma única solução para  $b$ , desde que a matriz  $X$  seja de posto completo. Desta forma, obtemos um vetor de resíduos homogêneo formado por zeros, ou seja:

$$e = y - Xb = 0 \quad (2.13)$$

- $m < n$ : este é o caso mais comum que nos deparamos na aplicação prática de problemas de MLR, onde existem mais amostras do que variáveis. Este fato não permite obter uma solução exata para a matriz de coeficientes  $b$ , mas sim uma solução que minimize o vetor de resíduos  $e = y - Xb$ , a partir da solução para o método de mínimos quadrados:

$$b = (X^T X)^{-1} X^T y \quad (2.14)$$

A avaliação destes 3 casos possíveis alerta para a necessidade de termos pelo menos tantas amostras quanto variáveis, o que pode ser obtido aumentando a amostragem ou deletando variáveis, de modo a sempre existir a inversa da matriz  $X^T X$  e evitar o fenômeno de colinearidade (GELADI; KOWALSKI, 1986).

### 2.2.2 Gradient Boosting Regressor

O *Gradient Boosting* constrói modelos aditivos de regressão ajustando os parâmetros de uma simples função (aprendiz básico ou do inglês *base learner*) pelos atuais “pseudo” - resíduos obtidos por mínimos quadrados a cada iteração. Os pseudo-resíduos são, portanto, o gradiente da função de perda sendo minimizados com relação aos valores do modelo a cada ponto dos dados de treinamento, avaliados na etapa atual (FRIEDMAN, 1999). Em um

problema de estimação de funções, temos um sistema constituído por um conjunto de variáveis aleatórias de saída chamadas de  $y$  e um conjunto de variáveis de entrada denominadas  $x = \{x_1, \dots, x_n\}$ . Dada uma amostra de treinamento  $\{y_i, x_i\}_1^N$  de valores  $(y, x)$ , o objetivo é encontrar uma função  $F^*(x)$  que estime  $y$  através dos valores de  $x$ , sendo que ao longo de toda a distribuição de valores  $(y, x)$  o valor de alguma função de perda  $\psi(y, F(x))$  é minimizado (FRIEDMAN, 1999), ou seja,

$$F^*(x) = \arg \min_{F(x)} E_{y,x} \psi(y, F(x)). \quad (2.15)$$

*Boosting* aproxima  $F^*(x)$  através de uma expansão por adição da forma

$$F(x) = \sum_{m=0}^M \beta_m h(x; a_m), \quad (2.16)$$

Onde a função  $h(x; a_m)$  é a *base learner* normalmente representada por funções simples de  $x$  com parâmetros  $a = \{a_1, a_2, \dots\}$ . Os coeficientes de expansão  $\{\beta_m\}_0^M$  e os parâmetros  $\{a_m\}_0^M$  são ajustados em conjunto aos dados de treinamento a cada passo. O processo se inicia com um chute inicial  $F_0(x)$  e então segue para  $m = 1, 2, \dots, M$ , ou seja,

$$(\beta_m, a_m) = \arg \min_{\beta, a} \sum_{i=1}^N \psi(y_i, F_{m-1}(x_i) + \beta h(x_i; a)) \quad (2.17)$$

E ainda,

$$F_m(x) = F_{m-1}(x) + \beta_m h(x; a_m) \quad (2.18)$$

O modelo *Gradient Boosting* aproximadamente resolve a equação 2.17, através de funções de perda arbitrárias (mas diferenciáveis)  $\psi(y, F(x))$  em um procedimento de duas etapas. Inicialmente a função  $h(x; a)$  é ajustada por mínimos quadrados

$$a_m = \arg \min_{a, \rho} \sum_{i=1}^N [\tilde{y}_{im} - \rho h(x_i; a)]^2 \quad (2.19)$$

Para os atuais pseudo-resíduos

$$\tilde{y}_{im} = - \left[ \frac{\partial \psi(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} \quad (2.20)$$

Então, obtido  $h(x; a_m)$ , o valor ótimo do coeficiente  $\beta_m$  é determinado

$$\beta_m = \arg \min_{\beta} \sum_{i=1}^N \psi(y_i, F_{m-1}(x_i) + \beta h(x_i; a_m)). \quad (2.21)$$

Esta estratégia substitui um potencialmente difícil problema de otimização da função 2.17 por outro baseado em mínimo quadrados (2.19) seguido de uma otimização de um único parâmetro (2.21) baseado no critério geral de perda  $\psi$  (FRIEDMAN, 1999). O *Gradient Boosting* trata este caso de uma maneira particular onde a função *base learner*  $h(x; a)$  é uma árvore de regressão de nó terminal-L. A cada iteração  $m$ , uma árvore de regressão particiona o espaço  $x$  em L regiões separadas  $\{R_{lm}\}_{l=1}^L$  e prediz uma constante para cada região

$$h(x; \{R_{lm}\}_1^L) = \sum_{l=1}^L \bar{y}_{lm} 1(x \in R_{lm}). \quad (2.22)$$

Aqui  $\bar{y}_{lm} = \text{média}_{x_i \in R_{lm}}(\tilde{y}_{lm})$  é a média de (2.20) em cada região  $R_{lm}$ . Os parâmetros da função *base learner* são as variáveis de divisão e correspondem aos pontos de divisão que definem a árvore, que por sua vez correspondem as regiões de partição na  $m$ -ésima iteração. Como a árvore prediz um valor constante para  $\bar{y}_{lm}$  em cada região  $R_{lm}$ , a solução para a equação (2.21) se reduz à seguinte forma simplificada:

$$\gamma_{lm} = \arg \min_{\gamma} \sum_{x_i \in R_{lm}} \psi(y_i, F_{m-1}(x_i) + \gamma). \quad (2.23)$$

A aproximação atual  $F_{m-1}(x)$  é atualizada em cada região correspondente:

$$F_m(x) = F_{m-1}(x) + v \cdot \gamma_{lm} 1(x \in R_{lm}). \quad (2.24)$$

O parâmetro de redução  $0 < v \leq 1$  controla a taxa de aprendizado do processo. Segundo (FRIEDMAN, 1999), foi demonstrado empiricamente que valores pequenos ( $v \leq 0,1$ ) levam a melhores generalizações dos erros. No trabalho de (FRIEDMAN, 1999) foi desenvolvido o algoritmo descrito para diversos critérios de perda: mínimos quadrados  $\psi(y, F) = (y - F)^2$ , desvio mínimo absoluto  $\psi(y, F) = |y - F|$ , Huber  $\psi(y, F) = (y - F)^2 1(|y - F| \leq \delta) + 2\delta(|y - F| - \delta/2) 1(|y - F| > \delta)$  e para classificação, log-verossimilhança negativa multinominal de classe  $K$ .

Para fins de comparação foi utilizada também a função *XGBRegressor* do pacote *XGBoost* (abreviação de *Extreme Gradient Boosting*) que utiliza o mesmo algoritmo do *Gradient Boosting* com outras variações de funções para composição do modelo.

### 2.3 Estratégia de Adaptação

Apesar de todo o esforço empregado na obtenção de um modelo que represente de forma adequada a variável que se procura estimar, devido à natureza dos processos industriais, os modelos tendem a perder aderência com o passar do tempo, ou seja, o erro entre o valor predito e o valor real tende a aumentar. Isto pode ser explicado por diversos motivos como, por exemplo: troca no fornecedor ou alteração nas características das matérias-primas, desgaste ou acúmulo de sujeira nos equipamentos, falhas em válvulas ou instrumentos entre inúmeros outros. Desta forma, foram desenvolvidos métodos para adaptar os modelos e corrigir os desvios existentes nos valores de predição para que o modelo elaborado permaneça útil. Estes métodos utilizam como referência os valores de análises de laboratório da propriedade estimada para a realização das adaptações e algumas das estratégias existentes serão abordadas a seguir:

- **Correção de Bias:** consiste na adição sobre o valor de predição em um instante de tempo  $t$  de uma correção obtida a partir da diferença existente entre o valor real e o valor predito neste mesmo instante. Geralmente este valor real é obtido a partir de análises de laboratório e a diferença citada é denominada de *bias*. Esta estratégia de correção pode ser equacionada da seguinte forma:

$$\hat{y}(t) = \hat{y}_p(t) + y_{BIAS} \quad (2.25)$$

Onde  $\hat{y}(t)$  é o valor final do modelo após a correção,  $\hat{y}_p(t)$  é o valor predito pelo modelo e  $y_{BIAS}$  é a correção calculada a partir da diferença entre o valor de predição no instante

$k, \hat{y}(k)$ , e valor de laboratório  $y_{LAB}(k)$  multiplicada por um fator de escala  $\alpha$  que está relacionada com a confiança da medida laboratorial.

$$y_{BIAS} = \alpha(y_{LAB}(k) - \hat{y}(k)) \quad (2.26)$$

Em processos industriais a frequência de análises não é elevada, podendo variar de 1 vez ao dia a 2 vezes por turno (ou 6 vezes ao dia) dependendo da necessidade. Isto gera a principal limitação e fonte de erro deste método, pois será adicionado um termo constante entre as análises, de maneira que o erro será carregado até a próxima análise. Por outro lado, a sua principal vantagem é a praticidade e facilidade de implementação (FACCHIN, 2005).

- **Mínimos quadrados recursivos:** este método realiza a atualização dos parâmetros do modelo e não a correção do valor predito como método anterior. Ele pode ser utilizado para modelos do tipo PLS e atualiza as matrizes de variância das entradas  $X^T.X$  e de covariância entre as entradas e saídas  $X^T.Y$  (CONZ, 2005). A abordagem matemática do algoritmo deste método pode ser consultada no trabalho de (TSAKIRIS, 2010).
- **Filtro de Kalman:** é um método versátil, pois difere erros do modelo e erros de medição, permitindo a adaptação do modelo a cada análise de laboratório realizada (CONZ, 2005). A abordagem matemática completa deste método pode ser consultada em (WELCH; BISHOP, 2006).

### 3 Metodologia Proposta

O desenvolvimento do analisador virtual para predição da variável Viscosidade Mooney do processo de fabricação de borracha do tipo SBR foi realizado baseado nos três pilares discutidos nas seções anteriores. A escolha inicial dos dados de processo a serem coletados foi realizada a partir do conhecimento teórico e prático do autor deste trabalho, que possui uma experiência de 8 anos como operador de processo da área de reação da unidade estudada. A aquisição dos dados de processo foi realizada através da extração dos registros de sensores de processo disponíveis a partir do software supervisor de produção – o PI Processbook do fabricante OSIsoft®, onde é possível visualizar as variáveis de processo em tempo real. A base de dados foi extraída utilizando o plug-in da ferramenta para Microsoft Excel, denominado de PI DataLink, com um intervalo de tempo de cinco minutos entre cada amostragem. Os dados de laboratório foram coletados a partir dos registros dos resultados de análise nas planilhas de controle utilizadas pela equipe de operação. Foi realizada uma pesquisa prévia da operação da planta nos seis primeiros meses de 2021 e foram eliminados os períodos de parada e partida da unidade para que o conjunto de dados fosse representativo aos períodos de operação em regime estacionário. O algoritmo utilizado para o desenvolvimento do analisador virtual foi implementado utilizando programação em python através do Google Colaboratory com a versão Python 3.7.12. O código completo utilizado pode ser encontrado no apêndice deste trabalho.

#### 3.1 Seleção de Variáveis

A etapa de seleção de variáveis é a etapa mais importante para que possamos obter um modelo de predição adequado. Para facilitar a compreensão do algoritmo adotado, a



metodologia aplicada nesta etapa será dividida em tópicos organizados na mesma sequência da implementação.

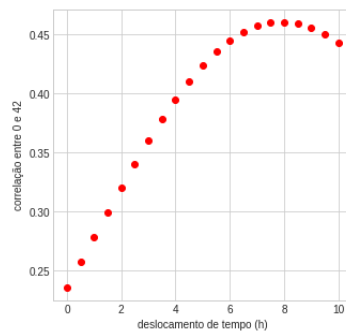
### 3.1.1 Primeira Inspeção dos Dados

Após carregada a planilha de dados do *Microsoft Excel* para o *Google Colaboratory* foi realizada uma criteriosa inspeção nos dados disponíveis com o intuito de identificar outliers, ou seja, valores espúrios originados de falhas dos sensores, por exemplo. Para identificação visual dos outliers foram plotadas as séries temporais de cada variável e, posteriormente, à identificação, foram removidos os dados correspondentes ao instante de tempo relativo à cada outlier. Para remoção destes outliers foram impostas restrições de valores para as variáveis de modo a manter apenas valores com significado físico e, através da função *drop* da biblioteca *Pandas*, foram removidas as linhas que contivessem valores fora do intervalo determinado. Com relação às propriedades de laboratório, elas são obtidas com uma frequência muito menor (a cada 4 horas) em relação aos demais dados de variáveis secundárias de processo (a cada 5 minutos) porém para evitar um erro gerado a partir da utilização de valores interpolados de viscosidade Mooney, para avaliar a qualidade de predição do modelo foram utilizados apenas os pontos correspondentes aos horários de coleta das amostras para comparação entre o valor de predição e o valor obtido através da análise laboratorial.

### 3.1.2 Deslocamento temporal de variáveis

Devido a característica do processo algumas variáveis não se relacionam com a variável a ser predita em instantes de tempo correspondentes, mas sim o seu efeito só fornecerá impacto na variável de resposta após determinado intervalo de tempo correspondente ao tempo de residência dos equipamentos a montante do ponto de coleta. Para exemplificar, podemos utilizar uma vazão de um insumo que é adicionado continuamente ao montante de uma cadeia de 10 reatores CSTR dispostos em série. Levando em consideração que o produto obtido da reação nestes reatores é coletado para análise de laboratório no final do último reator da série, o valor da vazão deste insumo adicionado em um instante de tempo  $t$  só impactará a especificação do produto no instante  $t + \tau$  onde  $\tau$  corresponde ao tempo de residência da mistura reacional na cadeia de reatores. Como o processo avaliado neste trabalho possui uma cadeia de reatores em série, foi necessário realizar um tratamento nos dados de vazões de entrada dos insumos nos reatores e nas temperaturas dos reatores, de modo a obter um deslocamento de tempo ótimo que corresponda a real correlação da variável secundária com a variável a ser estimada. A alternativa encontrada para realizar esta análise foi plotar o valor do coeficiente de correlação de Pearson entre a variável secundária e a variável a ser estimada para deslocamentos de tempo crescentes até o valor máximo do tempo de residência possível e visualizar o ponto onde obtemos o maior valor absoluto deste coeficiente de correlação. O deslocamento ótimo de tempo para cada variável foi definido como o valor de deslocamento de tempo em que foi obtido o maior valor absoluto do coeficiente de correlação de Pearson. Abaixo temos o exemplo de um dos gráficos utilizados para realizar esta avaliação.

Figura 3: Estudo do deslocamento de tempo ótimo



Fonte: O Autor.

Após realizada esta análise, foi verificado se os deslocamentos de tempo ótimos tinham significado físico plausível e foram implementados estes deslocamentos através da função *shift* do *Pandas*.

### 3.1.3 Determinação da dimensionalidade do Problema e Seleção das Variáveis Secundárias

Para determinação da dimensionalidade, ou seja, da quantidade de variáveis necessárias para representar o conjunto de dados, foi realizado uma análise de componentes principais (PCA) e plotando o percentual da variância explicada à medida que foram adicionados mais componentes. Primeiramente, utilizando o conjunto completo de dados, foram escalonadas as variáveis através do *StandardScaler* da função *preprocessing* da biblioteca *scikit-learn* de modo a obter variáveis cujo a média é igual a 0 e o desvio padrão é igual a 1. Este pré-processamento visa eliminar os efeitos de grandes diferenças de escala entre as variáveis. Em seguida foi utilizada a função PCA da biblioteca *scikit-learn* e o seu atributo *explained\_variance\_ratio* para quantificar o percentual da variância representado por determinado número de componentes. A partir desta metodologia, foi possível obter uma noção da quantidade de variáveis que seriam necessárias para compor o modelo de predição.

Visando selecionar as variáveis verdadeiramente correlacionadas com a característica a ser predita foi utilizada a regressão de Lasso em conjunto com o pacote *Lars*, plotando o coeficiente de determinação ( $R^2$ ) do modelo gerado a partir da variação do parâmetro de ajuste da força da penalidade no método Lasso. A ideia foi obter a menor quantidade de variável possível (referenciado pelo valor obtido no PCA) que mantivesse um valor de  $R^2$  próximo ao valor máximo obtido com o número total de variáveis.

A partir da análise do gráfico gerado, foi verificado o valor do fator de penalidade (chamado aqui de  $\alpha$ ) de modo a não ocorrer queda brusca no  $R^2$ , o qual deveria estar entre  $1 \times 10^{-3}$  e  $1 \times 10^{-2}$ , resultado este que será detalhado na seção de resultados. A regressão de Lasso ajustada com a regressão de ângulos mínimos *Lars* foi implementada através da função *LassoLars* da biblioteca *scikit-learn*.

## 3.2 Elaboração do Modelo de Inferência

A etapa de elaboração do modelo de inferência constituiu na implementação de diferentes técnicas de análise multivariável cujos parâmetros foram determinados a partir

do ajuste a um conjunto de dados de calibração do modelo fornecido. Posteriormente, os modelos gerados foram utilizados para predição da variável de interesse a partir do fornecimento de variáveis secundárias de um novo conjunto de dados de teste e as métricas de avaliação dos modelos foram verificadas a fim de identificar o modelo que apresentou a melhor capacidade de predição. A elaboração do modelo de inferência pode ser categorizada por etapas conforme segue:

- a) Segmentar o conjunto de dados em dois grupos: um grupo de dados de treinamento ou calibração do modelo, cujo a função é obter o melhor ajuste de coeficientes de forma a minimizar o vetor de resíduos gerado a partir dos valores de  $x$  e  $y$  que constituem os dados de treinamento e outro grupo de dados de teste, onde ao ajuste do modelo obtido a partir dos dados de treinamento será utilizado na predição dos valores de  $y$  dados os valores de  $x$  que compõem o grupo de teste, o que chamaremos de  $y_{pred}$ ;
- b) Calibrando os parâmetros de diferentes modelos de análise multivariável a partir dos dados de treinamento;
- c) Utilizando os modelos ajustados na etapa c), foram calculados valores de  $y$  sendo fornecidos novos valores de  $x$  pertencentes ao grupo de teste;
- d) Calculadas as métricas para avaliação dos modelos  $R^2$ , MAE, MSE e  $exp\_variance$  a partir dos valores de  $y$  reais que compõem os dados de teste,  $y_{test}$ , e dos valores de  $y_{pred}$ .
- e) Foram removidas sucessivamente as variáveis com menor coeficiente da regressão Lasso até que foram obtidos os melhores valores dos índices calculados na etapa e). Além dos índices de avaliação dos modelos, foram plotados os valores de  $y_{test}$  e  $y_{pred}$  de modo a visualizar o modelo que apresentou a melhor resposta e os pontos onde ocorreram os maiores erros;

### 3.3 Estratégia de Adaptação

Para elaborar uma correção preliminar do modelo, foi realizado, em um primeiro momento, o cálculo dos resíduos, ou seja, os valores de erro entre  $y_{test}$  e  $y_{pred}$  a partir dos resultados obtidos na etapa e) da seção anterior. Os valores de resíduo foram utilizados como valor de  $y$ , ou seja, o valor a ser estimado para obtenção de um modelo para predição dos valores de resíduo e então somar estes valores preditos ao valor de  $y$  predito para a variável de interesse. De posse dos valores de resíduo, foi aplicado o mesmo algoritmo descrito nas seções 3.1 e 3.2 para obtenção dos modelos de resíduo.

De posse dos modelos de predição da variável principal e dos resíduos, foi implementada uma estratégia de correção do modelo a partir de valores de laboratório. Devido a praticidade, a estratégia escolhida foi a correção de bias. De modo a minimizar o efeito de erros de análise e do intervalo de tempo, sem novas atualizações do valor de bias, foi utilizada para o fator de correção uma proporção de 70% do valor atual do bias e 30% do valor oriundo da atualização a partir do erro entre o valor de análise e o valor predito pelo modelo, sendo este o melhor ajuste obtido a partir do método de tentativa e erro.

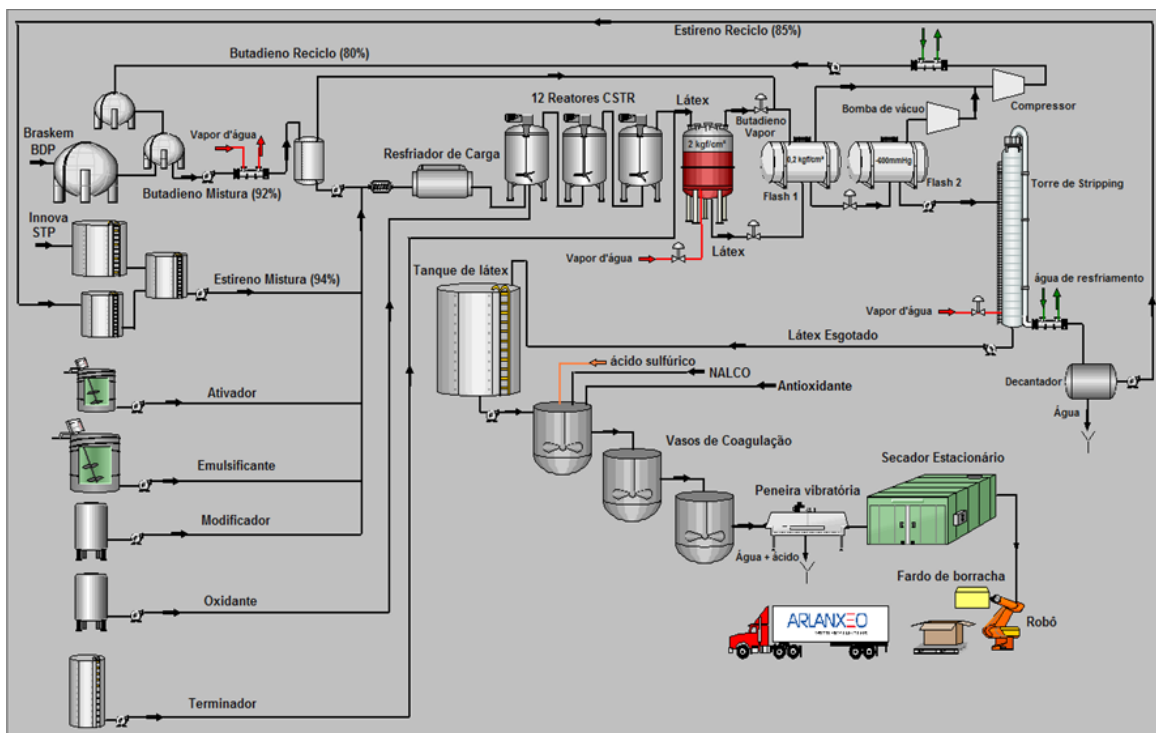
## 4 Estudo de Caso

A planta objeto deste estudo produz borracha do tipo SBR (*Butadiene and Styrene Rubber*), ou seja, borracha de butadieno estireno obtida através de um processo de polimerização a frio, via radicais livres. A principal característica a ser monitorada para avaliar a especificação da borracha é a viscosidade Mooney, que é uma medida da resistência do elastômero mediante a aplicação de uma força de cisalhamento dentro do regime elástico e está intimamente relacionada com as condições de processamento desta borracha para obter utensílios úteis como, por exemplo, o pneu, através de um processo de vulcanização. Vale salientar que a vulcanização não é realizada dentro da planta estudada, porém a viscosidade Mooney é a principal variável exigida pelos clientes que irão comprar esta borracha para realizar este processo. Por este motivo a viscosidade Mooney será a propriedade que buscaremos estimar através do analisador virtual aqui desenvolvido.

### 4.1 Descrição da Unidade de Polimerização de borracha SBR

É importante destacar que o autor do presente texto possui 8 anos de experiência como operador técnico da área de reação da planta descrita neste capítulo, portanto boa parte das informações aqui registradas são provenientes do conhecimento prático do dia a dia e outras são retiradas dos manuais de operação internos da empresa onde a fábrica fica instalada, sendo que estes manuais não podem ser utilizados para consulta pública. Para contextualizar a explicação da unidade que virá a seguir é representada na Figura 4 o processo de produção de borracha SBR de forma simplificada.

Figura 4: Desenho ilustrativo do processo de fabricação de borracha sintética



Fonte: O Autor.

Como pode ser observado na Figura 4, o processo inicia por uma área de armazenamento de matérias-primas onde são recebidos os monômeros que constituem a borracha: o 1,3-butadieno e o estireno com elevada pureza (mínimo 99,5%). Os demais insumos utilizados na área de polimerização são transferidos por bombas centrífugas para a área de polimerização (ou reação). Para que ocorra a uniformização da temperatura de todos os fluxos antes de sua entrada na cadeia de reatores, eles passam por um trocador de calor do tipo casco e tubo onde ocorre a refrigeração dos fluxos de entrada na reação até a temperatura de 12°C. A refrigeração no resfriador de carga e nos reatores é realizada através da vaporização de amônia líquida, e para isso existe uma área auxiliar de refrigeração por amônia, onde ocorre a compressão e condensação da amônia vaporizada na reação para a sua reutilização como fluido refrigerante. As variáveis de processo de área de refrigeração por amônia são importantes para estimar a taxa de calor que está sendo liberada na reação, visto que ela é exotérmica.

A reação é composta por uma série de até 15 reatores CSTR (*Continuous Stirred-Tank Reactor*) e cada reator possui um controle de temperatura através de um conjunto de feixes tubulares contendo amônia líquida que ficam submersos na mistura reacional no interior do reator, cuja amônia líquida vaporiza através da transferência do calor liberado na reação em calor latente de vaporização da amônia, mantendo a temperatura do reator dentro de uma faixa de especificação de 8 a 12 °C. A mistura reacional é denominada de látex e a reação é conduzida até uma conversão de 66% (valores maiores elevam a formação de ligações cruzadas entre as macromoléculas de polímero e tornam a borracha mais quebradiça e de difícil processamento) e, portanto, sobra uma quantidade de 34% de monômeros não reagidos (butadieno e estireno) frente a quantidade dosada que precisam ser removidos do látex na área de Recuperação de Monômeros para que possam ser reutilizados no processo.

Antes de ingressar na área de recuperação de monômeros, o látex passa por um vaso de acúmulo onde é aquecido até 40°C por meio da injeção de vapor d'água vivo para início do desprendimento do butadieno não reagido do látex. Do vaso de acúmulo saem duas tubulações para a área de recuperação de monômeros: a tubulação de fundo que contém látex e a tubulação de topo, também chamada de linha pulmão, que contém butadieno vapor e realiza o controle de pressão do final da cadeia de reatores. Ao ingressar na área de recuperação de monômeros, o látex passa por dois vasos de flash, onde no topo sai o butadieno vapor removido. O primeiro vaso de flash opera com pressão positiva de 0,2 kgf/cm<sup>2</sup> e o butadieno removido na saída de vapores é succionado por compressores centrífugos e alternativos para que seja pressurizado e posteriormente liquefeito em condensadores do tipo casco e tubo com água de resfriamento para então ser bombeado novamente até a área de armazenamento. O segundo vaso de flash é uma complementação na recuperação de butadieno e opera a pressão manométrica negativa de -600 mmHg, sendo que o butadieno removido na saída de vapores é succionado por bombas de vácuo centrífugas e alternativas que descarregam na sucção dos compressores acima citados e a partir daí realiza o mesmo fluxo já descrito. O látex livre de butadieno não reagido é bombeado para uma torre de *stripping* que opera com pressão negativa e onde é injetado vapor d'água em contracorrente para borbulhamento no látex e arraste do estireno não-reagido. A coluna é formada por pratos perfurados para aumentar a área de contato entre látex e vapor d'água, sendo o látex alimentado no topo da coluna, o vapor é alimentado na base da coluna e o látex desce para os pratos inferiores, transbordando pelos *downcomers*. No tubulão de topo da coluna, uma mistura de vapores de estireno e água passa por

condensadores que operam com água de resfriamento nos tubos e então a mistura é enviada para um decantador, onde ocorre a separação de fases, ficando o estireno sobrenadante e a água com valor residual de estireno dentro dos parâmetros ambientais é drenada para o rejeito orgânico. O estireno sobrenadante transborda em um vertedouro presente no interior do decantador e é então bombeado para a área de armazenamento de monômeros para ser reutilizado no processo. O látex de fundo da coluna é denominado de látex esgotado (livre de monômeros não-reagidos) e é bombeado para os tanques de armazenamento de látex para a alimentação da coagulação.

Na área de coagulação, o látex recebe ácido sulfúrico para quebra de estabilidade das moléculas de polímero na solução coloidal (látex), ocorrendo a coagulação da borracha e formando duas fases distintas: grumos de borracha (sólida) e uma solução de água ácida de pH 4,0 que parte é realimentada na coagulação e parte é drenada para o rejeito orgânico. Além do ácido sulfúrico, também são dosados nos vasos de coagulação o antioxidante (para retardar o envelhecimento da borracha quando estocada) e o auxiliar de coagulação, para manter a uniformidade no tamanho dos grumos de borracha obtidos na coagulação. Após os vasos de coagulação, a corrente de borracha e água ácida é encaminhada para uma peneira vibratória, onde há a separação entre a borracha e a água de coagulação. Após a peneira, os grumos de borracha são encaminhados para uma desumidificadora, que é uma espécie de extrusora, para a continuação da secagem dos grumos e, na sequência estes são enviados por meio de um transportador pneumático até um secador estacionário para a finalização da secagem até índices de material volátil de no máximo 0,5%. O secador estacionário é formado por quatro zonas de secagem e uma zona de resfriamento, sendo que cada zona de secagem possui queimadores próprios para o fornecimento de ar quente, através da queima do gás natural. Depois do secador, os grumos passam novamente por um desintegrador para separação dos grumos aglomerados na camada de borracha formada no secador e os grumos são encaminhados por meio de um elevador de canecas para uma balança onde são separadas as frações com um peso de até 35 kg. Estes grumos então são liberados da balança para uma prensa hidráulica para a formação dos fardos de borracha que são embalados e colocados em caixas metálicas através de um robô para que sejam vendidos aos clientes.

O enfoque deste trabalho é a elaboração de um analisador virtual para estimar a viscosidade Mooney da borracha, medida esta que é obtida através de análise laboratorial no látex coletado na saída da cadeia de reatores, a partir dos dados de variáveis de processo secundárias que possuem correlação com tal características do elastômero. Do ponto de vista prático o monitoramento contínuo da viscosidade Mooney pode trazer algumas vantagens, visto que atualmente o látex proveniente da área de recuperação de monômeros é armazenado em diversos tanques. O látex encaminhado para a coagulação é oriundo de um tanque de mistura, onde são misturadas quantidades de látex vindos dos tanques de armazenamento após estes serem analisados em laboratório. A operação é realizada desta forma devido à falta de um controle rígido dos valores de viscosidade Mooney obtidos na reação. Este fato faz com que o látex esfrie ao ficar muito tempo parado nos tanques de armazenamento e como a coagulação deve ocorrer a 70°C se faz necessário o uso de uma quantidade maior de vapor d'água para atingir esta condição de processo. Dentre os benefícios que podem ser obtidos a partir de implementação do analisador virtual podemos citar:

- Controle mais rígido das especificações obtidas na reação, visto que hoje as análises são realizadas a cada 4 horas e possíveis correções por parte do operador de processo é realizada somente após a obtenção de cada análise;
- Obtenção de um produto mais uniforme e de mais qualidade, visto que a variação de viscosidade Mooney do polímero será menor e reduzirá a largura da dispersão do peso molecular do elastômero;
- Economia de vapor d'água devido a possibilidade de enviar o látex diretamente da torre de *stripping* para a coagulação, de modo que se reduza a perda de temperatura antes de ingressar na coagulação (a temperatura do fundo da coluna de *stripping* fica na faixa de 70 a 80°C).

Na próxima seção se descrevem as variáveis medidas em linha, ou seja, as variáveis secundárias de processo e as variáveis medidas em laboratório, as quais desejamos estimar a partir do modelo a ser elaborado.

## 4.2 Variáveis medidas em linha

No capítulo 4.1 foi realizada uma explanação sobre a unidade de fabricação de borracha SBR como um todo, porém a partir deste ponto o foco será voltado para as áreas de reação e recuperação de monômeros. Abaixo constam duas tabelas, a Tabela 1 com a descrição dos equipamentos que compõem estes setores e a Tabela 2 com a listagem as variáveis que compõem o conjunto inicial dos dados de processo coletados.

**Tabela 1:** Descrição dos equipamentos de processo.

TAG	Descrição
TC-01	Trocador de calor do tipo casco e tubo, também chamado de resfriador de carga da reação. Nos tubos passa a corrente de alimentação da reação e no casco ocorre a vaporização da amônia líquida utilizada para refrigerar esta corrente até a temperatura de 12°C.
R-00 ao R-14	Reatores de Polimerização do tipo CSTR. Operam sempre cheios e possuem feixes internos tubulares onde há amônia líquida que é vaporizada para manter o controle de temperatura nos reatores.
Ponto de Coleta	Ponto de amostragem de látex na saída da reação. O látex amostrado já está misturado com o terminador e é "ventado" antes da coleta, onde é submetido a pressão negativa por 10 minutos para remoção do butadieno não reagido.
V-01	Vaso de acúmulo da saída da reação. Neste vaso ocorre injeção de vapor vido no látex para aquecê-lo até a temperatura de 40°C e iniciar o desprendimento do butadieno não reagido para ser recuperado na área de recuperação de monômeros. O controle de pressão deste vaso é realizado a partir da corrente de butadieno vapor do topo deste vaso.
V-02	Vaso de expansão a pressão. Vaso de flash onde ocorre grande parte da vaporização do butadieno não reagido.
V-03	Vaso de expansão a vácuo. Vaso de flash onde ocorre a finalização da vaporização do butadieno não reagido.
C-01	Compressor centrífugo acionado por turbina a vapor para compressão dos vapores de butadieno oriundos do V-02.
C-02	Bomba de vácuo centrífuga acionada por turbina a vapor para compressão dos vapores de butadieno oriundos do V-03.
B-01	Bomba centrífuga para bombeio de látex do V-03 para a torre de <i>stripping</i>
T-02	Torre de <i>stripping</i> com bandejas perfuradas para remoção do estireno não reagido a partir de arraste com vapor d'água vivo.
B-02	Bomba centrífuga para bombeio de látex da T-02 para os tanques de armazenamento.



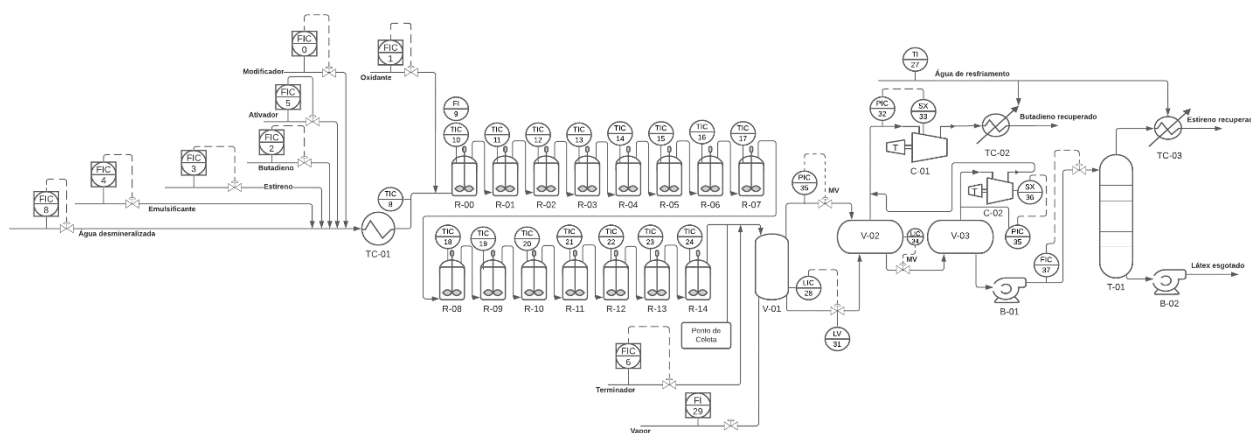
**Tabela 2:** Descrição das variáveis de processo

TAG	Nome	Descrição	Unidade
FIC-0	Vazão de modificador para a reação	Indicação da vazão mássica de modificador para a alimentação da reação obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
FIC-1	Vazão de oxidante para a reação	Indicação da vazão mássica de oxidante para a alimentação da reação obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
FIC-2	Vazão de butadieno para a reação	Indicação da vazão mássica de butadieno para a alimentação da reação obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
FIC-3	Vazão de estireno para a reação	Indicação da vazão mássica de estireno para a alimentação da reação obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
FIC-4	Vazão de emulsificante para a reação	Indicação da vazão mássica de emulsificante para a alimentação da reação obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
FIC-5	Vazão de ativador para a reação	Indicação da vazão mássica de ativador para a alimentação da reação obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
FIC-6	Vazão de terminador para a reação	Indicação da vazão mássica de terminador para a alimentação da reação obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
FIC-7	Vazão de água desmineralizada para a reação	Indicação da vazão mássica de água desmineralizada para a alimentação da reação obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
TIC-8	Temperatura de saída do resfriador de carga	Indicação de temperatura da corrente de alimentação da reação na saída do resfriador de carga obtido a partir de um sensor de temperatura do tipo PT-100.	°C
FI-9	Vazão de amônia vaporizada no primeiro reator	Indicação da vazão de amônia vaporizada no primeiro reator de polimerização devido ao calor de reação, obtida a partir de um medidor de vazão do tipo placa de orifício. Não há controle desta variável, é apenas uma indicação.	kg/h
TIC-10 ao TIC-24	Temperatura do Reator 0 até a temperatura do Reator 14	Indicação de temperatura dos reatores de polimerização obtido a partir de um sensor de temperatura do tipo PT-100.	°C
PIC-25	Pressão da amônia vapor	Pressão do vaso de sucção dos compressores da área de refrigeração de amônia, controlado a partir da capacidade utilizada dos compressores desta área e obtido a partir de um sensor de pressão capacitivo	kgf/cm <sup>2</sup>
SX-26	Capacidade compressores	Capacidade utilizada dos compressores da área de refrigeração de amônia. Um compressor equivale a 100% e meio compressor equivale a 50% de capacidade.	%
TI-27	Temperatura da água de resfriamento	Indicação de temperatura da água de resfriamento utilizada nos condensadores da fábrica obtida a partir de um sensor de temperatura do tipo PT-100.	°C
LIC-28	Nível do vaso de acúmulo	Indicação de nível do vaso de acúmulo da cadeia de reatores, obtido a partir de um sensor de pressão capacitivo.	%
FI-29	Vazão de vapor para o vaso de acúmulo	Indicação da vazão de vapor para os vasos de acúmulo para vaporização do butadieno que não reagiu, obtido a partir de um medidor de vazão do tipo placa de orifício.	°C
PIC-30.MV	Abertura de válvula da fase vapor do vaso de acúmulo	Indicação da abertura da válvula de controle de pressão do vaso de acúmulo. Está diretamente relacionada com a vazão da fase vapor que sai do topo do vaso de acúmulo.	%
LV-31	Abertura de válvula de saída de látex do vaso de acúmulo	Indicação da abertura da válvula de controle de nível do vaso de acúmulo. Está diretamente relacionada com a vazão de látex enviada para a área de reação.	%
PIC-32	Pressão de sucção dos compressores da área de recuperação de monômeros	Indicação da pressão de sucção dos compressores da área de recuperação de monômeros obtida a partir de um sensor de pressão capacitivo. Esta variável é controlada a partir da capacidade utilizada dos compressores da área.	kgf/cm <sup>2</sup>
SX-33	Capacidade compressores	Capacidade utilizada dos compressores da área de recuperação de monômeros. Cada compressor operando na sua capacidade máxima equivale a 100%.	%
LIC-34.MV	Abertura de válvula da saída de látex do vaso de expansão a pressão	Indicação da abertura da válvula de controle de nível do vaso de expansão a pressão. Está diretamente relacionada com a vazão de látex enviada para o vaso de expansão à vácuo.	%
PIC-35	Pressão de sucção das bombas de vácuo da área de recuperação de monômeros	Indicação da pressão de sucção das bombas de vácuo da área de recuperação de monômeros obtida a partir de um sensor de pressão capacitivo. Esta variável é controlada a partir da capacidade utilizada das bombas de vácuo da área.	mmHg
SX-36	Capacidade bombas de vácuo	Capacidade utilizada das bombas de vácuo da área de recuperação de monômeros. Cada compressor operando na sua capacidade máxima equivale a 100%.	%
FIC-37	Vazão de látex para a torre de stripping	Indicação da vazão de látex enviado do vaso de expansão à vácuo para a torre de stripping, obtido a partir de um medidor de vazão mássico do tipo coriólis.	kg/h
38	Número de reatores	Contagem do número de reatores de polimerização que estão em operação.	reatores
39	Vazão da polimerização	Vazão de reação setada pelo operador para atender a demanda de produção e utilizada também para ajuste do teor de sólidos totais para manter o valor dentro da especificação desejada.	t/d
40	Taxa de reação	Vazão da reação dividido pelo número de reatores em operação.	t/(d.reator)
41	Razão Bd/St	Razão entre as vazões dos monômeros butadieno e estireno.	
42	Viscosidade Mooney	Resultado da análise laboratorial da viscosidade mooney do polímero obtido no final da cadeia de reatores. É uma das variáveis a serem estimadas pelo analisador virtual. Esta variável está diretamente relacionada com a distribuição de peso molecular do polímero obtido.	N.A.
43	Sólidos totais	Teor de sólidos totais do látex ao final da cadeia de reatores obtido a partir de análise de laboratório. Esta variável está diretamente relacionada com a conversão de monômeros na reação.	%
44	Estireno Combinado	Teor de estireno combinado do polímero obtido no final da cadeia de reatores obtido a partir de análise de laboratório.	%

O objetivo do analisador virtual desenvolvido neste trabalho é estimar a característica analisada em laboratório na amostra de látex coletada no final da cadeia de reatores (no local indicado na figura 5 como “Ponto de Coleta”), a Viscosidade Mooney. A ideia é a utilização de variáveis secundárias de processo medidas em linha para obtenção de um modelo que nos permita estimar as análises de laboratório. Para a escolha inicial das variáveis cujo dados seriam coletados para a elaboração do modelo foi utilizado o conhecimento prévio do processo obtido pelo autor e a partir deste conjunto inicial de dados foi aplicado o algoritmo descrito no capítulo 3.1 para identificação das variáveis correlacionadas com as características a serem estimadas. Na figura 5 pode-se observar o fluxograma de processo das áreas de reação e recuperação de monômeros com o TAG de identificação de cada uma das variáveis de processo cujos dados foram coletados para o presente estudo.



Figura 5: Fluxograma de Processo da unidade de polimerização



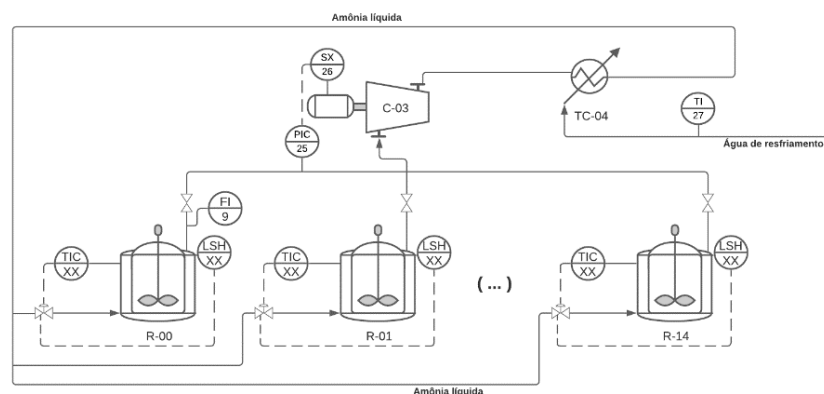
Fonte: O Autor.

Observando a Figura 5 podem ser identificadas as variáveis “FIC-xx” que representam indicações de vazão oriundas de transmissores instalados nos medidores de vazão mássico do tipo coriólis dos insumos dosados na reação. Para maiores informações sobre os insumos que são dosados na reação e sobre o seu papel no processo polimerização consulte o texto disponível no Apêndice A deste trabalho. Esta corrente de alimentação é então encaminhada para o TC-01 que é o resfriador de carga da reação e que refrigera este fluxo até a temperatura de 12°C. Esta temperatura é controlada através do TIC-8, que compara a indicação do sensor de temperatura instalado na saída do resfriador de carga com o *setpoint* e realiza o ajuste necessário na pressão de vaporização da amônia neste trocador, ou seja, a pressão da amônia opera em cascata com a temperatura de saída e para redução da temperatura se faz necessária a redução da pressão de amônia e vice-versa. O controle de temperatura dos reatores de polimerização é representado pelos TIC-10 ao TIC-24 e é realizado em automático com a válvula de injeção de amônia líquida para os feixes dos reatores. A variável manipulada deste controle é o tempo em que a válvula permanece aberta na forma de um bias no tempo pré-estabelecido, ou seja, para redução da temperatura do reator haverá aumento do tempo que a válvula permanecerá aberta dentro deste intervalo de tempo. Na prática o que ocorrerá é uma variação de nível dos feixes de amônia líquida, que causará uma modificação na área de troca térmica utilizada para refrigeração do látex contido no reator. A pressão de vaporização da amônia líquida é mantida dentro de um *range* setado pelo operador através da modulação de capacidade dos compressores alternativos de duplo estágio e duplo efeito da área de refrigeração, sendo que para ocorrer a elevação da pressão o compressor que estiver em automático modula para 50% da capacidade, onde apenas um dos efeitos de compressão permanecerá operante. O valor da pressão de amônia vapor necessária é determinado pela demanda de controle de temperatura dos reatores, sendo mantida sempre a maior pressão possível em que não ocorra descontrole na temperatura dos reatores para que se reduza o consumo de energia elétrica. Posteriormente a cadeia de reatores, o látex já misturado ao terminador da reação é transferido para o vaso de acúmulo onde recebe injeção de vapor vivo para início da vaporização do butadieno não reagido. A injeção de vapor é controlada por uma válvula de controle que opera em automático com a temperatura do vaso de acúmulo, porém a variável utilizada para o conjunto de dados é a indicação de vazão do vapor utilizado, sendo

que esta variável não é controlada e sim uma consequência da necessidade de aquecimento do conteúdo do vaso de acúmulo. A pressão do vaso de acúmulo é controlada a partir de uma válvula de controle na saída de butadieno vapor no topo deste vaso, sendo esta pressão uma variável importante para que se consiga a transferência do látex do interior do vaso de acúmulo para o vaso de expansão a pressão que ocorre por diferencial de pressão. No vaso de expansão a pressão é delimitado o início da área de recuperação de monômeros e a sua pressão é controlada a partir da capacidade utilizada dos compressores de butadieno, representada para variável SX-33, onde na prática ocorre o aumento de rotação dos compressores para que a pressão seja reduzida. A finalização da recuperação de butadieno é realizada em um segundo vaso de flash, o vaso de expansão a vácuo, cujo a pressão é controlada de maneira similar à do vaso de expansão a pressão, ocorrendo elevação da capacidade (ou rotação) das bombas de vácuo para aumento de nível de pressão negativa no interior do vaso, o que é representada pela variável SX-36. Após removido o butadieno não reagido a recuperação do estireno não reagido é realizado na torre de *stripping* T-01, sendo a única variável coletada desta parte do processo a vazão de alimentação de látex para esta coluna.

Na Figura 6 pode-se observar uma representação simplificada do sistema de controle de temperatura dos reatores de polimerização. Este sistema está representado de forma genérica pois é idêntico para cada um dos quinze reatores que compõem a cadeia de reação, sendo os compressores e condensadores de amônia constituintes da área auxiliar de refrigeração por amônia que recebe a amônia vaporizada pelo calor exotérmico de reação de todos os reatores e realiza a sua compressão e condensação para que a amônia líquida seja redirecionada para os reatores, fechando assim o ciclo de refrigeração.

Figura 6: Fluxograma simplificado do sistema de refrigeração dos reatores



Fonte: O Autor.

Ainda sobre as variáveis secundárias, pode ser observada na Tabela 2 a variável 38, que representa o número de reatores que estão em operação e que é ajustada conforme a necessidade de vazão do processo e a disponibilidade de equipamentos livres de intervenção da manutenção, a variável 39 que representa a vazão de polimerização setada pelo operador e que determinará os *setpoints* de vazão dos insumos dosados na reação e que é ajustada

para enquadramento dos valores de sólidos totais dentro da faixa de especificação (para aumento do teor de sólidos se reduz a vazão e vice-versa), a variável 40 que representa a taxa de reação que nada mais é do que a divisão da variável 39 pela variável 38 e a variável 41, que é a divisão da variável FIC-2 pela variável FIC-3. Foram inseridas ainda ao conjunto de dados outras três variáveis calculadas a partir dos dados de temperatura já descritos: a variável 45 ( $Var_{45}$ ) que é a média de temperaturas do R-00 ao R-04, a variável 46 que é a média de temperaturas do R-05 ao R-09 e a variável 47 que analogamente é a média de temperaturas do R-10 ao R-15.

$$Var_{45} = \frac{(TIC_{10} + TIC_{11} + TIC_{12} + TIC_{13} + TIC_{14})}{5} \quad (4.1)$$

$$Var_{46} = \frac{(TIC_{15} + TIC_{16} + TIC_{17} + TIC_{18} + TIC_{19})}{5} \quad (4.2)$$

$$Var_{47} = \frac{(TIC_{20} + TIC_{21} + TIC_{22} + TIC_{23} + TIC_{24})}{5} \quad (4.3)$$

O objetivo destas três novas variáveis criadas é capturar informações relativas ao impacto da temperatura na viscosidade Mooney do elastômero e reduzir o efeito negativo sobre o modelo de temperaturas de reatores que porventura sejam retirados de operação, o que resultariam em outliers devido a temperatura fora dos limites de controle e sem gerarem impacto real na variável principal.

### 4.3 Variável medida em laboratório: Viscosidade Mooney (VM)

Os elastômeros, por apresentarem tanto características de materiais sólidos, como de líquidos, apresentam uma resposta reológica viscoelástica, quando submetidos a tensões de deformação, o que implica em uma combinação de respostas elásticas e viscosas. Esse comportamento reológico possui grande importância para definir o desempenho e o processamento dos artefatos de borracha e isto explica a importância do acompanhamento destas características. Em outras palavras, pode-se dizer que os polímeros apresentam um comportamento intermediário entre um sólido elástico ideal que segue a Lei de Hooke e um fluido viscoso ideal considerado um fluido Newtoniano. Este fato é observado pois materiais poliméricos sofrem deformação instantânea, quando submetidos a tensões de cisalhamento (Lei de Hooke), porém este fenômeno é seguido de uma deformação contínua ou fluência (também denominada de “creep”), o que faz com que parte da deformação seja recuperada quando cessada a atuação desta tensão e outra parte seja recuperada ao longo de determinado tempo, podendo até mesmo se tornar uma deformação permanente em alguns casos.

Consequente, durante o processamento de polímeros, estes são submetidos a diferentes tipos de deformação em decorrência da geometria complexa do maquinário utilizado. Para a modelagem deste comportamento é necessário assumirmos a sobreposição de pelo menos dois regimes reológicos simples: um fluxo de cisalhamento simples, como por exemplo no interior de uma extrusora em um processo de extrusão e outro fluxo de estiramento ou alongamento simples, no início da matriz neste mesmo processo de extrusão por exemplo (GUERRA; FURTADO; COUTINHO, 2004)

No processo abordado neste trabalho, o ensaio realizado para definir a excelência do produto visa caracterizar condições de processabilidade e vulcanização para o consumidor. Neste contexto é utilizado o teste para definir a plasticidade do polímero baseado na taxa de cisalhamento para a determinação da Viscosidade Mooney, ensaio este que é padronizado através na norma ASTM D-1646 e acabou se tornando o ensaio padrão utilizado na indústria de polímeros. O equipamento utilizado neste ensaio é denominado de viscosímetro Mooney, desenvolvido por Melvin Mooney e este determina a viscosidade do elastômero em função de sua resistência ao cisalhamento. A viscosidade do polímero pode ser definida a partir da seguinte equação:

$$\eta = K.(PM)^a \quad (4.4)$$

Onde:

$\eta$  = viscosidade intrínseca

$K$  = constante de proporcionalidade dependente do tipo de elastômero

$PM$  = massa molar média do polímero

$a$  = grau de enovelamento da molécula

As constantes “ $K$ ” e “ $a$ ” podem ser obtidas experimentalmente, sendo que dependem de fatores relacionados a forma das moléculas de polímero, como por exemplo da quantidade de ramificações presentes no elastômero. Pode ser observado que quanto maior o peso molecular médio do polímero, maior a viscosidade e, portanto, menor será a sua plasticidade (GUERRA; FURTADO; COUTINHO, 2004). O viscosímetro Mooney é formado por um rotor metálico que gira com uma velocidade de 2 rpm, sendo que existem dois rotores padrões para o ensaio: o rotor grande (L) para polímeros de baixa viscosidade e o rotor pequeno (S) para o polímero de alta viscosidade. No processo em questão é utilizado apenas o rotor do padrão “L”. A temperatura do ensaio é conduzida a 100°C e o corpo-de prova (amostra de borracha) consiste em duas partes do material a ser testado, com diâmetros e espessuras suficientes para preencher toda a cavidade do equipamento. As duas partes do corpo-de-prova envolvem o rotor de metal, que gira por determinado período. A resistência à rotação é medida em unidades arbitrárias como Viscosidade Mooney (VM) e é apresentada conforme a expressão 4.2 (GUERRA; FURTADO; COUTINHO, 2004):

$$VM = 50 - ML 1 + 4 (100^\circ C) \quad (4.5)$$

Onde:

50 – Indica unidade de viscosidade

M – Indica Mooney

L – Indica que foi usado o rotor grande

4 – Indica o tempo de ensaio em minutos

(100°C) – Indica a temperatura do ensaio

No controle do processo de produção de borracha SBR a viscosidade Mooney é controlada através do pcm (partes por cem de monômero) do agente de transferência de cadeia (modificador) adicionado, sendo que quanto maior o pcm menor será a viscosidade Mooney e vice-versa. As correções de Mooney inerentes às necessidades de processo são realizadas pelos operadores de processo alterando-se o pcm de modificador de modo a atingir a especificação desejada do produto.

## 5 Resultados e Discussões

Seguindo o mesmo padrão adotado até aqui, os resultados serão expostos na sequência em que os 3 pilares do desenvolvimento de analisadores virtuais foram executados.

### 5.1 Variáveis Escolhidas

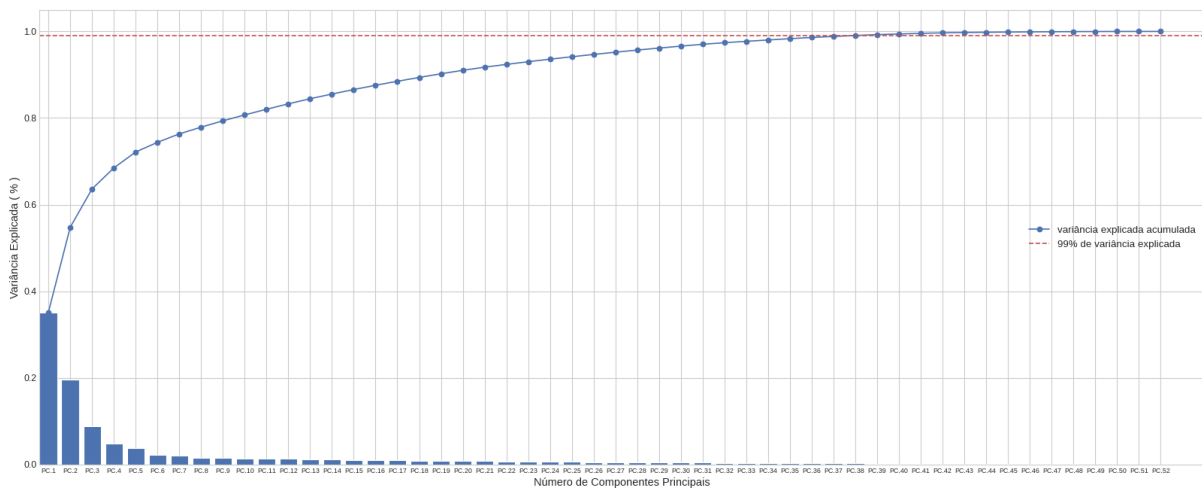
Quanto aos deslocamentos de tempo ótimos, os resultados obtidos foram coerentes com a prática, ou seja, o maior deslocamento de tempo foi obtido nas vazões de insumos de alimentação da reação dosados a montante da cadeia de reatores (com exceção do FIC-6 pois o terminador é dosado no final da cadeia de reatores) e o deslocamento de tempo ótimo para a temperatura dos reatores veio decrescendo à medida que se aproximavam do final da cadeia. Na Figura 3 da seção 3.1.2 temos um exemplo do gráfico utilizado para avaliar de forma qualitativa o deslocamento de tempo ótimo da variável FIC-0, sendo o valor de máxima correlação obtido com um deslocamento de tempo de 7 horas. A Tabela 3 apresenta os valores de deslocamento de tempo ótimos obtidos para cada variável.

**Tabela 3:** Deslocamento de tempo ótimo para as variáveis do conjunto de dados com localização a montante do ponto de coleta

Variável	Deslocamento de tempo ótimo (h)	Variável	Deslocamento de tempo ótimo (h)	Variável	Deslocamento de tempo ótimo (h)
FIC-0	7	TIC-10	6	TIC-20	4
FIC-1	7	TIC-11	6	TIC-21	3
FIC-2	7	TIC-12	6	TIC-22	2,5
FIC-3	7	TIC-13	5,5	TIC-23	2,5
FIC-4	7	TIC-14	5	TIC-24	2
FIC-5	7	TIC-15	5	40	5
FIC-6	0	TIC-16	5		
FIC-7	7	TIC-17	5		
FIC-8	7	TIC-18	4,5		
FI-9	7	TIC-19	4,5		

Como citado na metodologia descrita na seção 3.1 para a seleção de variáveis primeiramente foi realizado um PCA no conjunto inicial de variáveis para obter uma noção da dimensionalidade do sistema.

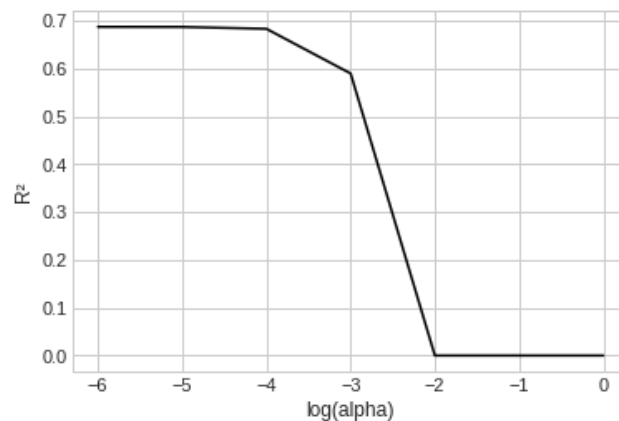
Figura 7: Variância explicada em função do número de principais componentes



Fonte: O Autor.

Podemos observar no gráfico acima que com 15 componentes podemos explicar aproximadamente 90% da variância dos dados, sendo possivelmente o restante da informação oriunda de ruídos. Por este motivo o número de variáveis selecionadas ficou próximo a este valor.

Em seguida foi aplicada Regressão de Lasso Lars para selecionar as variáveis com maior correlação com a viscosidade Mooney.

Figura 8: Análise do impacto do fator de penalidade no valor de  $R^2$ 

Fonte: O Autor.

Foi selecionado o número de variáveis correspondente a um valor de alpha (fator de penalidade) que mantivesse um valor de  $R^2$  aceitável, ou seja, o número mínimo de variáveis para que não iniciasse queda acentuada no valor do coeficiente de determinação ( $R^2$ ). Analisando a faixa intermediária com valor de alpha entre  $1 \times 10^{-3}$  e  $2 \times 10^{-3}$  se chegou a um valor intermediário de  $1,5 \times 10^{-3}$  com valor de  $R^2$  representado na Tabela 4 abaixo, correspondente a 14 variáveis com coeficiente de regressão diferente de zero que foram: FIC-0, FIC-2, FIC-4, FIC-5, FIC-7, FI-9, TIC-17, TIC-19, TIC-23, TIC-24, PIC-25, PIC-30.MV, 38, 39. A ideia do uso destes valores intermediários foi obter um conjunto com o menor número de

variáveis possível (representando as variáveis mais fortemente correlacionas) e que fosse capaz de carregar o máximo de informação, ou seja, sem apresentar uma grande queda no coeficiente de determinação.

**Tabela 4:** Valores de  $R^2$  e nº de variáveis com coeficiente de regressão diferente de zero em função do parâmetro de penalidade alpha

Alpha	$R^2$	Nº variáveis diferentes de 0
0,0010	0,589	17
0,0011	0,572	16
0,0012	0,554	15
0,0013	0,534	15
0,0014	0,513	15
<b>0,0015</b>	<b>0,490</b>	<b>14</b>
0,0016	0,466	13
0,0017	0,442	13
0,0018	0,417	11
0,0019	0,391	11

Depois de selecionadas as variáveis foi verificada a coerência quanto a escolha delas. Tendo em vista que a viscosidade Mooney é diretamente proporcional ao peso molecular do elastômero, pode-se esperar que fatores que influenciem a conversão da reação tenham correlação direta com este parâmetro. Desta forma as vazões selecionadas (FIC-0, FIC-2, FIC-4, FIC-5, FIC-7) são plausíveis de serem escolhidas pois impactam a reatividade da polimerização, sendo o destaque o FIC-0, que representa a vazão do insumo dosado para controlar o crescimento das cadeias de polímero (para maiores informações a respeito do mecanismo de polimerização consultar Apêndice A) sendo a variável mais fortemente correlacionada com a viscosidade Mooney. A variável FI-9 por ser uma medida da quantidade de amônia vaporizada no primeiro reator é uma medida indireta da reatividade no início da reação, visto que quanto maior o calor liberado (reação exotérmica) maior tende a ser a conversão final obtida o que impacta a viscosidade Mooney. Pelo mesmo motivo as temperaturas dos reatores selecionados TIC-17, TIC-19, TIC-23 e TIC-24 também possuem relação teórica com a viscosidade Mooney. A variável PIC-25 possivelmente foi selecionada pelo mesmo motivo da variável FI-9, pois quanto maior for a quantidade de amônia vaporizada nos reatores em decorrência do calor de reação maior será a pressão do vaso de sucção dos compressores de amônia. A variável PIC-30.MV, que corresponde a abertura de válvula da fase vapor do vaso de acúmulo, após a cadeia de reatores, possui sentido em ser escolhida, pois quanto maior a conversão, menor será a quantidade de butadieno que irá sobrar para ser recuperado e, portanto, menor deverá ser a abertura desta válvula para manter a pressão setada no controlador e vice-versa.

Esperava-se que mais temperaturas fossem selecionadas para comporem o modelo, visto que são 15 reatores no total e apenas duas temperaturas foram escolhidas. A solução encontrada foi a criação das variáveis 45, 46 e 47, correspondentes a média da temperatura dos reatores já comentadas na seção 4.2. A inserção de cada uma delas ao modelo será avaliada na próxima seção.

## 5.2 Modelos Finais Obtidos

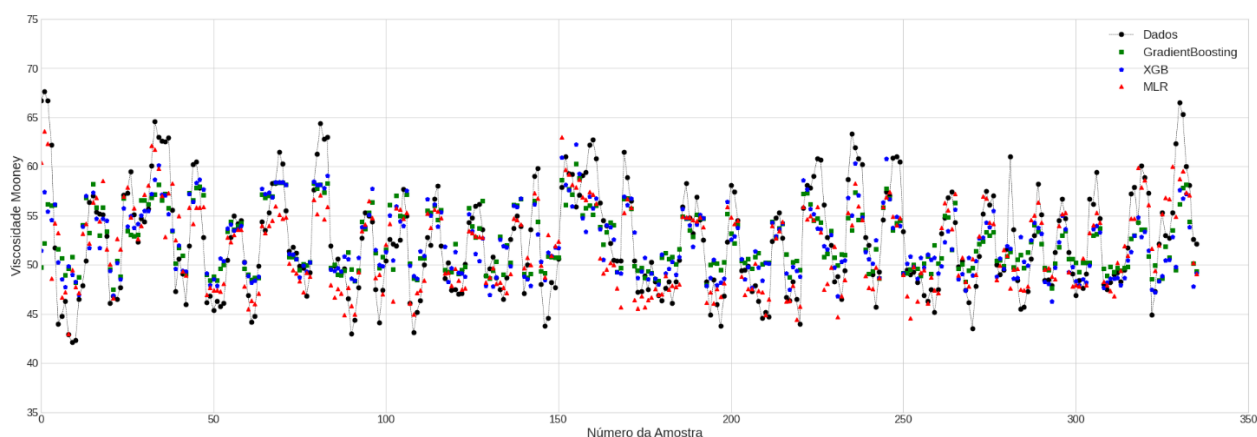
Escolhido o conjunto de variáveis, foram testadas algumas técnicas de modelagem multivariável: a regressão linear multivariável e dois modelos baseados no algoritmo de árvores de decisão, o *Gradient Boosting Regressor* e o *XGB Regressor* como alternativa de modelos não lineares para fins de comparação. Do total de 336 amostras que compunham o conjunto de dados cujos pontos foram abrangidos por análises de laboratório, 71% foram destinadas para o grupo de dados de treinamento ou calibração do modelo e os 29% restantes foram destinadas para o grupo de dados de teste. Estes valores foram ajustados de modo a obter um grupo de dados de teste o mais extenso possível para testar a capacidade de predição do modelo gerado, sem utilizar uma técnica específica. Os resultados de predição de viscosidade Mooney para o conjunto de teste dos 3 modelos acima citados calibrados com o grupo de dados de treinamento estão expostos a seguir na tabela 5.

**Tabela 5:** Resultados de predição para os três modelos testados

Métrica de avaliação	MLR	Gradient Boosting	XGB
$R^2$	0,635	0,511	0,546
<i>Explained variance</i>	0,650	0,511	0,548
<i>MAE</i>	2,495	2,958	2,878
<i>MSE</i>	10,780	14,443	13,405
<i>RMSE</i>	3,283	3,800	3,539
<i>Tempo de Processamento (s)</i>	0,0052	13,189	2,597

Analisando os resultados de comparação entre as métricas obtidos podemos destacar uma clara vantagem do Modelo de Regressão Linear frente aos outros dois modelos baseados em algoritmos de árvores de decisão, pois além de apresentar valores de erro inferiores, apresentou, também, um tempo de processamento pelo menos 500 vezes menor frente aos demais. Para contribuir na análise do melhor modelo a ser utilizado nas etapas seguintes de elaboração do analisador virtual foi plotado um gráfico comparando os três modelos obtidos. Os dados foram conectados por uma fina linha para facilitar a visualização dos pontos correspondentes na sequência, sendo que estes não são contínuos.

**Figura 9:** Comparação da predição para o conjunto de dados de teste entre os três modelos obtidos: MLR, *Gradient Boosting* e *XGBRegressor*



Fonte: O Autor.



Como podemos observar na Figura 9 acima, o modelo que apresentou melhor resposta frente as mudanças de alvo realizadas na variável de predição foi o modelo MLR chegando mais próximo aos valores extremos de viscosidade Mooney, principalmente para valores mais baixos desta variável. Com relação aos modelos não-lineares foi possível observar resposta frente as mudanças de alvo, porém sem a capacidade de se aproximar de forma adequada aos extremos. Tendo em vista a maior facilidade de implementação do modelo de MLR e dos melhores resultados obtidos nesta análise comparativa, este será o modelo utilizado para aplicação do restante da metodologia utilizada no desenvolvimento do analisador virtual.

Na tentativa de melhorar a resposta do modelo, foram realizadas duas tentativas de inclusão de novas variáveis:

- a) Inclusão de 3 novas variáveis (45,46 e 47) referentes as médias de temperaturas dos reatores conforme descrito no final da seção 4.2. Estas variáveis foram inseridas após a seleção de variáveis realizada pela Regressão de Lasso, para que o modelo considere de alguma forma o impacto da temperatura dos demais reatores que não foram selecionados previamente por não terem apresentado variação significativa no espaço de tempo estudado;
- b) Inclusão de 4 novas variáveis referentes ao termo  $1/T^2$  referente ao valor de aproximação em série de Taylor do segundo termo da equação para a equação de Arrhenius  $k = C_1 e^{-\frac{C_2}{T}}$  com o intuito de tentar capturar comportamento não lineares destas temperaturas.
- c) Inclusão de novas variáveis para simular a concentração dos insumos selecionados no primeiro reator da série. Estas variáveis foram calculadas através da média das últimas 50 vazões obtidas, ou seja, o que representa a média das últimas 4 horas para a variável FIC-xx correspondente. A estas variáveis foram atribuídas os números 52 (FIC-0), 55 (FIC-2), 56 (FIC-4), 57 (FIC-5), 58 (FIC-7). Também foram criadas variáveis dos últimos 50 valores médios para as variáveis FI-9 (59), TIC-17 (60), TIC-19 (61), TIC-23 (62) e TIC-24 (63). A ideia de utilizar estas variáveis é a de carregar informações das variáveis nos intervalos entre as análises. A melhor configuração possível utilizando estes dados foi mantendo apenas a variável FIC-0 e as temperaturas dos dois últimos reatores (TIC-23 e TIC-24), sendo as demais variáveis substituídas pelos valores médios. Os resultados obtidos constam na Tabela 6.

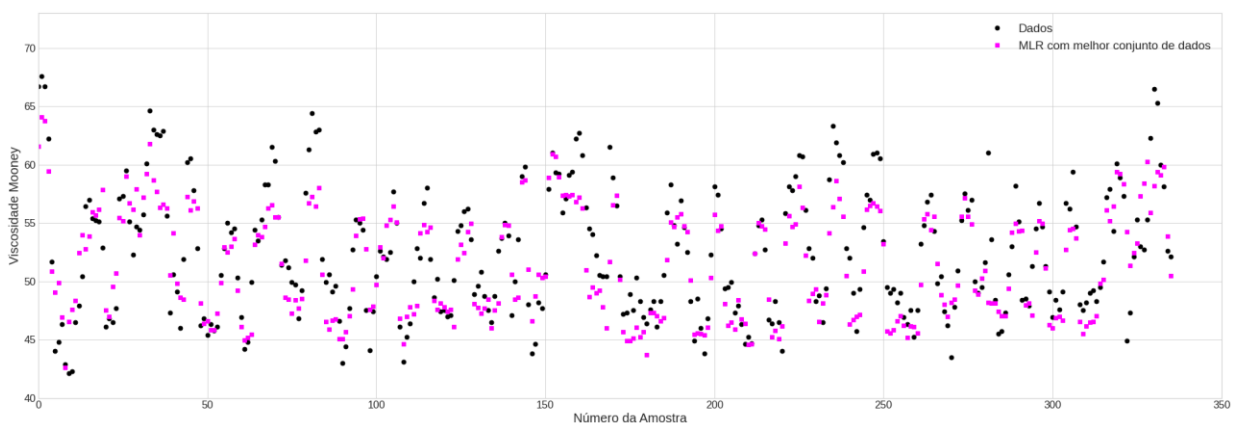
**Tabela 6:** Comparação de resultados para inclusão de variáveis

Métrica de avaliação	Conjunto original	Inclusão das Temp. médias	Inclusão $1/T^2$	Inclusão variáveis médias
$R^2$	0,635	0,643	0,631	0,707
<i>Explained variance</i>	0,650	0,672	0,631	0,741
<i>MAE</i>	2,495	2,448	2,538	2,283
<i>MSE</i>	10,780	10,564	10,903	8,652
<i>RMSE</i>	3,283	3,250	3,302	2,941

Com base nos resultados expostos na Tabela 6, pode-se observar que a inclusão da média das temperaturas não alterou muito os resultados perante o conjunto de dados inicial, causando uma leve melhora nos resultados. Como é importante considerarmos o impacto da temperatura dos demais reatores que compõem a série no analisador virtual estas variáveis serão mantidas no conjunto de dados e o modelo utilizado será o que seus parâmetros foram calibrados com base neste conjunto de dados. Já a inclusão de variáveis representativas aos termos  $1/T_2$  ocasionou uma pequena piora nos resultados e por isso, apesar de teoricamente esta tentativa ser válida, estas variáveis não serão adotadas no conjunto de dados para comporem o modelo.

Com relação as variáveis de média dos últimos 50 dados coletados, que visam simular o efeito da concentração dos insumos no primeiro reator e não somente da variável de vazão deste insumo, foi observado melhora significativa nos resultados provavelmente pelo fato de serem carregadas informações contidas nos intervalos entre duas análises, o que na prática impactam o resultado da viscosidade Mooney. Abaixo temos o gráfico do resultado da predição da viscosidade Mooney, a partir do modelo MLR com o melhor conjunto de dados definido.

Figura 10: Melhor predição para a Viscosidade Mooney



Fonte: O Autor.

Definido então o conjunto de variáveis e calibrado os parâmetros do modelo de MLR com ele, partiu-se então para o estudo de estratégias de adaptação para correção do modelo a partir das análises de laboratório.

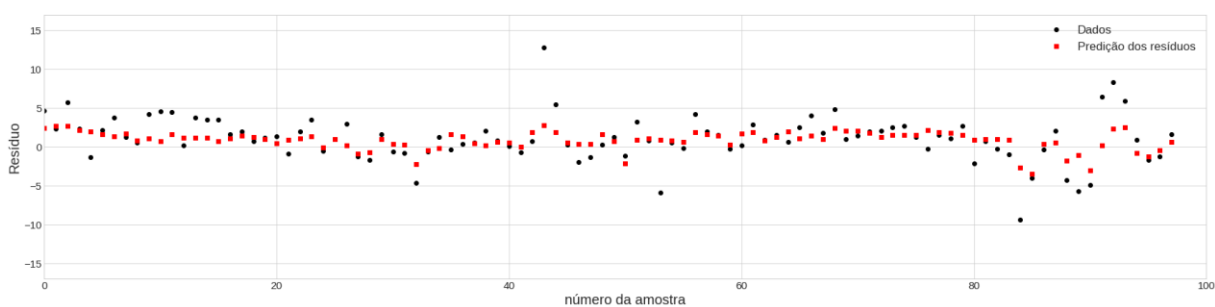
### 5.3 Estratégias de adaptação – modelo do resíduo

Para elaboração de uma estratégia de adaptação do modelo gerado, primeiramente foi repetido todo o procedimento até aqui utilizado para elaboração de um modelo para predição dos resíduos, de modo a avaliar quais as variáveis que são mais fortemente relacionadas com os erros de predição. A ideia foi obter um modelo simples, com no máximo 3 variáveis que estejam fortemente relacionadas com o erro de modo a buscar predições de viscosidade Mooney de maior qualidade.

Realizando a regressão de Lasso para o conjunto de dados inicial excluindo-se a variável de predição (variável 42, viscosidade Mooney) e tendo como valor y a ser predito os resíduos

calculados a partir do erro entre os dados de laboratório e os resultados de predição foi ajustado o fator de penalidade  $\alpha$  até que restassem apenas três variáveis com coeficiente de regressão diferente de zero, indicando que são estas as variáveis mais fortemente correlacionadas com o resíduo. As variáveis selecionadas foram: a variável 30 que é a abertura de válvula da fase vapor do vaso de acúmulo de látex, localizado na jusante a cadeia de reatores, a variável 43, que é o teor de sólidos totais e a 51 que é o inverso do quadrado da temperatura do último reator de série. Vale ressaltar que a escolha da variável 43 é interessante, pois trata-se de uma medida direta da conversão da reação e cujo análise é extremamente simples e pode ser executada em menos de 5 minutos, se necessário, sendo que, na prática, realmente é observado aumento de viscosidade Mooney quando ocorre aumento do teor de sólidos totais e vice-versa. Escolhidas as três variáveis para compor o modelo dos resíduos, foi dividido então o conjunto de dados em calibração e teste para ajuste dos parâmetros de modelo de MLR para os resíduos seguido do teste da sua capacidade de predição. O valor do coeficiente de determinação obtido para o modelo dos resíduos foi de  $R^2 = 0,387$  e o ajuste do modelo pode ser visualizado no gráfico a seguir.

Figura 11: Ajuste obtido para o modelo dos resíduos



Fonte: O Autor.

Apesar do ajuste obtido através dos modelos dos resíduos não ter excelente se observa uma tendência de ajuste aos valores reais através do uso das variáveis selecionadas. Como a proposta inicial era a obtenção de um modelo simples com o uso de poucas variáveis e como o modelo acompanha a tendência dos valores de resíduo se espera melhora nos resultados obtidos a partir da implementação deste modelo de correção.

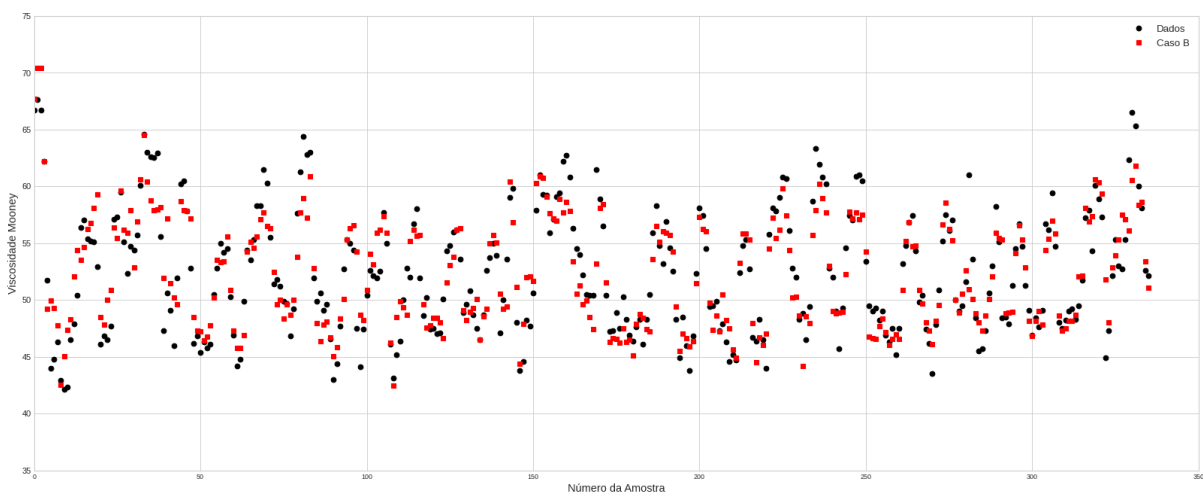
Após a soma entre os valores dos modelos de predição da variável 42 e do modelo dos resíduos obtivemos então os valores finais de predição. A estes novos valores não foi implementada a estratégia de correção de bias pois aqui trabalhamos apenas com os pontos correspondentes ao mesmo instante de tempo das análises de laboratório. Como a correção pelo bias deverá estar deslocada em 1 hora (tempo de duração da análise) com relação ao horário de coleta da amostra, a única opção seria deslocar a correção pelo bias em 4 horas, de modo que a análise anterior corrigisse a análise atual. Como não será este o método implementado, foi optado por aplicar a estratégia de correção do bias apenas no novo grupo de dados coletado, cujo resultados serão descritos no capítulo a seguir. Abaixo temos os resultados obtidos para as 2 situações citadas: uso somente do valor de predição para a variável 42 (chamado de Caso A) e uso dos modelos de predição da variável 42 e dos resíduos (Caso B).

**Tabela 7:** Resultados das estratégias de adaptação implementadas

Métrica de avaliação	Caso A	Caso B
$R^2$	0,707	0,798
<i>Explained variance</i>	0,741	0,798
<i>MAE</i>	2,283	1,881
<i>MSE</i>	8,652	5,978
<i>RMSE</i>	2,941	2,445

Fonte: O Autor.

Avaliando os resultados obtidos observar-se que a estratégia de implementação do modelo dos resíduos apresentou resultados satisfatórios, visto que o erro inerente a análise de viscosidade Mooney é de até 1,0 ponto para mais ou para menos, ou seja, para duas alíquotas de uma mesma amostra de borracha testada no mesmo viscosímetro é aceitável uma variação de até 1,0 ponto para o erro da análise. Abaixo temos a representação gráfica do ajuste obtido através da implementação da estratégia do Caso B para o conjunto de dados de teste.

**Figura 12:** Resultado da predição obtida através da estratégia do Caso B

Fonte: O Autor.

Como podemos observar no gráfico acima, o analisador virtual apresentou uma resposta que seria útil para seu uso no controle de processo para monitoramento contínuo dos valores de viscosidade Mooney produzidos no processo, com valores de erro que se enquadram tanto no erro inerente a análise de viscosidade Mooney quanto a faixa operacional de especificação do produto para os dois *grades* de borracha existentes:

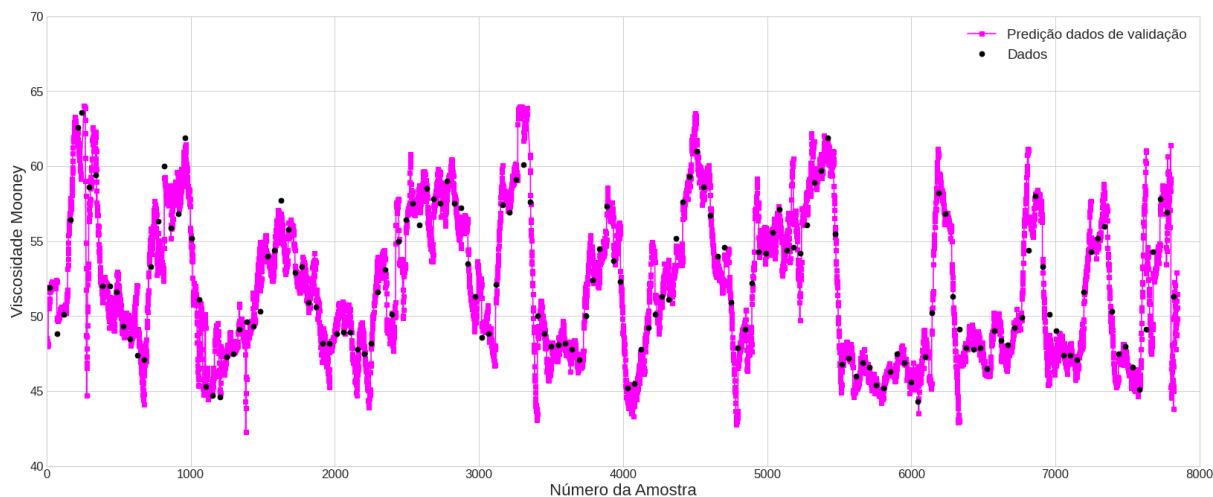
- Grade L (Baixa Viscosidade): Alvo de VM = 48 com limite inferior de controle de 46 e limite superior de controle de 50.
- Grade H (Alta Viscosidade): Alvo de VM = 52 com limite inferior de controle de 50 e limite superior de controle de 54.

O erro médio absoluto possui valor inferior ao delta de dois pontos de oscilação frente ao alvo de viscosidade Mooney implementada para cada um dos *grades* de produção, o que demonstra que a ferramenta desenvolvida no presente trabalho possui potencial como instrumento de monitoramento e controle de processo produtivo de borracha SBR. Entretanto, a análise de laboratório da viscosidade Mooney leva em torno de 1 hora para a sua realização e a atualização do bias deverá ocorrer com um atraso de uma hora com relação ao horário de coleta da amostra. Esta estratégia será discutida na próxima seção.

#### 5.4 Validação do modelo

Para simular a aplicação prática do modelo de predição e da estratégia de correção desenvolvidos foram coletados novos dados de processo referentes ao mês de agosto de 2021. Desta vez uma nova predição foi realizada, não somente para comparação com as análises de laboratório, mas, sim, a cada 5 minutos de modo a simular uma situação de disponibilidade de monitoramento contínuo da viscosidade Mooney durante a operação da planta. Para efeitos de comparação da qualidade da predição foram utilizados apenas os pontos correspondentes às análises laboratoriais. Vale destacar que nesta etapa foram utilizados os resultados de análise de laboratório deslocados no tempo em 1 hora para a correção do bias. Na figura 13 abaixo temos o gráfico correspondente a um período de predição de 15 dias da variável viscosidade Mooney durante o mês de agosto de 2021 com a estratégia desenvolvida neste trabalho.

Figura 13: Resultado de predição da viscosidade Mooney e comparação com resultados de laboratório



Fonte: O Autor.

O gráfico acima demonstra a potencialidade do método desenvolvido para o uso como ferramenta de monitoramento contínuo do valor de viscosidade Mooney no processo produtivo de borracha SBR, visto que as predições obtidas se aproximam de maneira satisfatória das análises de laboratório (pontos em vermelho).

Foi realizado ainda um estudo para avaliar a possibilidade da redução da frequência de análises de 2 vezes ao turno (ou a cada 4 horas) para 1 vez ao turno (ou a cada 8 horas). Esta ação proporcionaria economia dos reagentes utilizados para as análises e permitiria que os técnicos químicos encarregados da execução destas análises pudessem dedicar este tempo a

outras atividades. Abaixo temos uma tabela com o comparativo de resultados para o caso de análise a cada 4 horas e para o caso de análise a cada 8 horas.

**Tabela 8:** Comparativo dos resultados para diferentes frequências de análises

Métrica de avaliação	Análise a cada 4 h	Análise a cada 8 h
$R^2$	0,832	0,720
<i>Explained variance</i>	0,832	0,721
<i>MAE</i>	1,323	1,716
<i>MSE</i>	3,360	5,589
<i>RMSE</i>	1,833	2,364

Como demonstram os resultados obtidos, o analisador virtual desenvolvido se apresenta como uma útil ferramenta para controle do processo, visto que os resultados obtidos apresentaram valores de erro dentro dos limites aceitáveis, possibilitando até mesmo manter um monitoramento da viscosidade Mooney com determinada precisão para frequências de análises menores que as habituais.

Analisando o resultado demonstrado na figura 13, observamos que o erro do modelo tende a diminuir para períodos em que ocorre estabilidade nos valores de viscosidade Mooney. Este fato reforça o argumento favorável à operação enviando o látex obtido na reação diretamente para a coagulação, ao invés de serem estocados em tanques de armazenamento para então serem misturados e após coagulados. Para operar desta nova maneira deverá ser fixado o alvo de viscosidade Mooney na reação, por exemplo: ao invés de operar com alvos de viscosidade se alternando entre 48 e 58 para na média obter uma mistura com viscosidade de 53, seria fixado o alvo de viscosidade na reação diretamente em 53. Implementando esta forma de operação é esperado, baseado nos resultados obtidos neste estudo, que ocorra melhora na qualidade do modelo, devido a operação em faixas mais estáveis de viscosidade. É importante salientar que se for implementada de forma satisfatória esta maneira de operação geraria economia de vapor d'água utilizado no processo e, conseqüentemente, menores custos devido a aquisição desta utilidade pois o látex enviado diretamente do fundo da coluna de *stripping* perderia uma quantidade menor de energia térmica para o ambiente por não ficar armazenado em tanques de mistura por determinado período.

### 5.5 Comparação com a literatura existente

De acordo com os resultados disponíveis no trabalho de (PADMAVATHI *et al.*, 2005), para a modelagem de viscosidade Mooney em um processo de produção de borracha de polibutadieno o valor do coeficiente de determinação obtido foi de  $R^2 = 0,435$  para o modelo MLR e 0,454 utilizando um modelo de regressão logarítmico. Já utilizando as redes neurais desenvolvidas no seu trabalho os resultados foram muito melhores, com  $R^2 = 0,926$  para uma rede neuronal do tipo Fuzzy (FNN) e  $R^2 = 0,982$  para uma rede neuronal de regressão generalizada (GRNN). O modelo foi gerado a partir de onze variáveis selecionadas.

## 6 Conclusão e sugestões para trabalhos futuros

A partir do desenvolvimento do analisador virtual realizado neste trabalho, pode-se concluir que esta ferramenta funciona como uma excelente alternativa para monitoramento e controle de processos industriais, utilizando de maneira útil a vasta disponibilidade de dados de processo existente para inferir a especificação do produto fabricado. Apesar de existirem diversas técnicas mais sofisticadas para elaboração do modelo de inferência e estratégias de adaptação mais robustas, o modelo MLR para inferência da viscosidade Mooney e estratégia de correção do bias apresentaram resultados satisfatórios com grande utilidade prática. Optou-se por extrair o máximo de resultados a partir destas técnicas, devido a maior facilidade de implementação na indústria, sendo necessário somente a utilização do software Microsoft Excel conectado ao supervisor de produção PI Processbook, através do plug-in PI DataLink para que o modelo e a estratégia de adaptação possam ser utilizados para inferência da viscosidade durante a produção, sendo estes implementados diretamente no Excel.

Com relação à qualidade das predições obtidas, apesar de um erro médio absoluto de 1,3 pontos de viscosidade Mooney, o que representa 2,5% do valor médio de 52 para a viscosidade Mooney, ser aceitável devido à faixa de operação existente e ao erro de 1,0 ponto aceitável em decorrência do processo de análise laboratorial, percebeu-se uma demora na resposta do modelo a aderir mudanças de alvo, quando este é alterado de valores baixos para valores elevados e talvez modelos com características não-lineares fossem capazes de reproduzir este comportamento. É importante salientar que em intervalos de operação onde se mantém o alvo de viscosidade por maiores intervalos de tempo a qualidade de predição do modelo tende a ficar consideravelmente maior, o que fornece uma ferramenta poderosa para que seja testada e implementada a estratégia de operação, enviando o látex produzido diretamente para a coagulação ao invés de serem destinados para tanques de armazenamento. Esta estratégia proporcionará economia, devido ao menor consumo de vapor d'água, e um produto de melhor qualidade devido a maior uniformidade na distribuição de pesos moleculares das cadeias de polímero que constituem o produto.

Como sugestões de trabalhos futuros, podemos citar o teste de outros modelos de inferência não-lineares, com as redes neuronais, por exemplo, com o objetivo de detectar os comportamentos não-lineares do sistema que não puderam ser reproduzidos pelo analisador desenvolvido e a implementação de estratégias de adaptação do modelo mais robustas, como o Filtro de Kalman Estendido, para que não haja correção não somente do valor da predição, a partir das análises de laboratório, mas também dos parâmetros do modelo. Destaca-se, ainda, como sugestão o estudo da qualidade de predição do analisador desenvolvido operando de maneira a enviar o látex produzido diretamente para a área de coagulação e, se for necessário, realizar novamente o algoritmo proposto neste trabalho, a fim de selecionar novas variáveis para composição do modelo e realizar um novo ajuste dos parâmetros dos modelos através destes dados, pois as condições de operação mudariam consideravelmente e talvez o modelo aqui desenvolvido não seja capaz de reproduzir esta nova condição operacional de maneira satisfatória.

## REFERÊNCIAS

- ABDI, Hervé; WILLIAMS, Lynne J. Principal component analysis. **Wiley Interdisciplinary Reviews: Computational Statistics**, [s. l.], v. 2, n. 4, p. 433–459, 2010. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1002/wics.101>. Acesso at: 18 Oct. 2021.
- BROWNE, Michael W. Cross-Validation Methods. **Journal of Mathematical Psychology**, [s. l.], v. 44, n. 1, p. 108–132, 2000. Disponível em: Acesso at: 17 Oct. 2021.
- CONZ, Vanessa. **Desenvolvimento de Analisadores Virtuais Aplicados a Colunas de Destilação Industriais**. [S. l.: s. n.], 2005. Disponível em: <https://lume.ufrgs.br/handle/10183/4985>. Acesso at: 23 Jun. 2021.
- FACCHIN, Samuel. **Técnicas de Análise Multivariável aplicadas ao Desenvolvimento de Analisadores Virtuais**. [S. l.: s. n.], 2005. Disponível em: <https://lume.ufrgs.br/handle/10183/8294>. Acesso at: 23 Jun. 2021.
- FONTI, Valeria. Feature selection using lasso. **Business Analytics**, [s. l.], 2017. Disponível em: [https://www.researchgate.net/profile/David-Booth-7/post/Regression-of-pairwise-trait-similarity-on-similarity-in-personal-attributes/attachment/5b18368d4cde260d15e3a4e3/AS%3A634606906785793%401528313485788/download/werkstuk-fonti\\_tcm235-836234.pdf](https://www.researchgate.net/profile/David-Booth-7/post/Regression-of-pairwise-trait-similarity-on-similarity-in-personal-attributes/attachment/5b18368d4cde260d15e3a4e3/AS%3A634606906785793%401528313485788/download/werkstuk-fonti_tcm235-836234.pdf). Acesso at: 15 Oct. 2021.
- FORTUNA, L. (Luigi). Soft sensors for monitoring and control of industrial processes. [s. l.], p. 270, 2007. Disponível em: Acesso at: 20 Aug. 2021.
- FORTUNA, Luigi *et al.* Soft Sensors for Monitoring and Control of Industrial Processes. **Soft Sensors for Monitoring and Control of Industrial Processes**, [s. l.], 2007. Disponível em: Acesso at: 20 Aug. 2021.
- FORTUNA, Luigi; GRAZIANI, Salvatore; XIBILIA, Maria Gabriella. Virtual instruments in refineries. **IEEE Instrumentation and Measurement Magazine**, [s. l.], v. 8, n. 4, p. 26–34, 2005. Disponível em: Acesso at: 20 Aug. 2021.
- FRIEDMAN, Jerome H. Stochastic Gradient Boosting. [s. l.], 1999. Disponível em: Acesso at: 30 Oct. 2021.
- GELADI, Paul; KOWALSKI, Bruce R. **PARTIAL LEAST-SQUARES REGRESSION: A TUTORIAL** *Analytica Chimica Acta*. [S. l.]: Elsevier Science Publishers B.V, 1986.
- GELADI, Paul; LINDERHOLM, Johan. Principal Component Analysis. **Comprehensive Chemometrics**, [s. l.], p. 17–37, 2020. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/B9780124095472148929>. Acesso at: 18 Oct. 2021.
- GUERRA, Bianca B.; FURTADO, Cristina R. G.; COUTINHO, Fernanda M. B. Avaliação reológica de elastômeros e suas composições. **Polímeros**, [s. l.], v. 14, n. 4, p. 289–294, 2004. Disponível em: Acesso at: 23 Jun. 2021.



- KVÆLSETH, Tarald O. Cautionary note about  $r^2$ . **American Statistician**, [s. l.], v. 39, n. 4, p. 279–285, 1985. Disponível em: Acesso at: 17 Oct. 2021.
- LENG, Chenlei; LIN, Yi; WAHBA, Grance. A note on the lasso and related procedures in model selection. **JSTOR**, [s. l.], 2006. Disponível em: <https://www.jstor.org/stable/24307787>. Acesso at: 15 Oct. 2021.
- LIU, Hongbin *et al.* Dynamic Nonlinear Partial Least Squares Modeling Using Gaussian Process Regression. **Industrial and Engineering Chemistry Research**, [s. l.], v. 58, n. 36, p. 16676–16686, 2019. Disponível em: Acesso at: 18 Oct. 2021.
- MARTINS PALHA JÚNIOR, José. **Otimização do processo de fabricação de borracha termoplástica para controle do auto-acoplamento**. 2003. - Universidade Federal de Pernambuco, [s. l.], 2003. Disponível em: <https://repositorio.ufpe.br/handle/123456789/6871>. Acesso at: 23 Jun. 2021.
- MENEGOLLA, Henrique Binotto. **ESTUDOS SOBRE A APLICAÇÃO DE AUTOENCODER PARA CONSTRUÇÃO DE INFERÊNCIAS NA INDÚSTRIA QUÍMICA**. 2019. [s. l.], 2019. Disponível em: <https://lume.ufrgs.br/handle/10183/205357>. Acesso at: 23 Jun. 2021.
- MOHR, Tiago. **Desenvolvimento de um Analisador Virtual para uma Planta de Polietileno de Alta Densidade**. [S. l.: s. n.], 2004. Disponível em: Acesso at: 23 Jun. 2021.
- MONTGOMERY; PECK. Introduction to Linear. **math.uh.edu**, [s. l.], 1982. Disponível em: [https://www.math.uh.edu/~tai/Book\\_Orders/2012\\_SPRING/MATH%205386\\_PETERS.pdf](https://www.math.uh.edu/~tai/Book_Orders/2012_SPRING/MATH%205386_PETERS.pdf). Acesso at: 17 Oct. 2021.
- PADMAVATHI, G. *et al.* Neural modelling of Mooney viscosity of polybutadiene rubber. **Computers and Chemical Engineering**, [s. l.], v. 29, n. 7, p. 1677–1685, 2005. Disponível em: Acesso at: 26 Nov. 2021.
- RANZAN, L. *et al.* Desenvolvimento de Inferências utilizando o Método de Otimização Colônia de Formigas. [s. l.], p. 12025–12032, 2015. Disponível em: Acesso at: 17 Oct. 2021.
- ROCHA, Fernanda Paula. UNIVERSIDADE ESTADUAL DE CAMPINAS Aplicação do Filtro de Kalman Estendido para estimação de populações em modelos epidemiológicos. [s. l.], 2019. Disponível em: <http://lattes.cnpq.br/1109424085323484>. Acesso at: 18 Oct. 2021.
- ROCHA, Tereza C.J.; SOARES, Bluma G.; COUTINHO, Fernanda M.B. The most important butadiene based elastomers employed in the automotive industry. **Polimeros**, [s. l.], v. 17, n. 4, p. 299–307, 2007. Disponível em: Acesso at: 16 Jun. 2021.
- SAYER, C.; LIMA, E.L.; PINTO, J.C. Dynamic modeling of SBR emulsion polymerization reactors refrigerated by thermosyphons. **Chemical Engineering Science**, [s. l.], v. 52, n. 3, p. 341–356, 1997.
- SMITH, Wendell v.; EWART, Roswell H. Kinetics of emulsion polymerization. **The Journal of Chemical Physics**, [s. l.], v. 16, n. 6, p. 592–599, 1948. Disponível em: <https://aip.scitation.org/doi/abs/10.1063/1.1746951>. Acesso at: 20 Jun. 2021.
- TSAKIRIS, Manolis. **ON THE REGULARIZATION OF THE RECURSIVE LEAST-SQUARES ALGORITHM**. [S. l.: s. n.], 2010.

- VAN DEN BERG. Introduction to Matlab + Mathematical aspects of bilinear factor models (PCA and PLS). [s. l.], 2003. Disponível em: Acesso at: 17 Oct. 2021.
- WELCH, Greg; BISHOP, Gary. **An Introduction to the Kalman Filter**. [S. l.: s. n.], 2006. Disponível em: [http://www.cs.unc.edu/~gb. .](http://www.cs.unc.edu/~gb.)
- WERLE, Leandro Osmar. **Analisadores virtuais baseados em modelo neural para monitoramento e controle de colunas de destilação com aquecimento distribuído**. [S. l.]: Florianópolis, 2012. Disponível em: <https://repositorio.ufsc.br/handle/123456789/99478>. Acesso at: 23 Jun. 2021.
- WISNIEWSKI, A.; ROHNELT, R. C. **A Prática da Concentração do Latex**. [S. l.: s. n.], 1947. Disponível em: Acesso at: 4 Jul. 2021.
- G. Padmavathi, M.G. Mandan, S.P. Mitra, K.K. Chaudhuri. **Neural modelling of Mooney viscosity of polybutadiene rubber**. Comput. Chem. Eng., 29 (2005), pp. 1677-1685

## APÊNDICE A

### A.1 Borracha de Estireno-Butadieno (SBR)

A borracha SBR é um copolímero formado a partir dos monômeros butadieno e estireno e é um dos mais consumidos polímeros sintéticos do mundo, sendo grande parte da produção destinada a aplicações na indústria automotiva principalmente como um dos componentes do material dos pneus (SAYER; LIMA; PINTO, 1997). Este tipo de polímero pode ser obtido por um processo em emulsão via radicais livre denominado E-SBR ou pelo processo em solução pela via aniônica denominado S-SBR, sendo o E-SBR o processo objeto deste estudo. O processo em emulsão pode ser conduzido a quente (em altas temperaturas) ou a frio (em baixas temperaturas), sendo que estas condições distintas de processo produzem elastômeros com características de processamento totalmente diferentes. A técnica de polimerização a frio é usualmente mais utilizada e são estas as condições de processo empregadas no processo aqui observado. O material obtido no final da etapa de polimerização a frio via radicais livres de borracha SBR é uma dispersão coloidal estável com aspecto semelhante ao leite de vaca denominada látex, que apesar de não ser o produto final (que são os fardos de borracha SBR) possui algumas aplicações comerciais, tais como tintas e revestimentos, sem que sejam necessárias operações de separação (ROCHA; SOARES; COUTINHO, 2007).

#### A.1.1 Polimerização em Emulsão a frio

A técnica de polimerização em emulsão a frio foi desenvolvida em meados de 1947, sendo a reação conduzida idealmente na temperatura de 5°C. A água é utilizada como meio dispersante da mistura reacional, sendo o principal inibidor desta mistura o oxigênio dissolvido que deve ser consumido a partir da adição de ditionito de sódio. Os demais componentes principais adicionados na polimerização de SBR são os monômeros butadieno e estireno, emulsificante e um sistema iniciador. O sistema iniciador utilizado neste processo trata-se de um sistema de oxidação-redução (redox) composto por um sal ferroso e um hidroperóxido (ROOH), onde os radicais livres formados são resultados da reação entre este hidroperóxido e íon ferroso que é oxidado de ferro II a ferro III e posteriormente regenerado através da ação de um sulfoxilato. Para o controle do peso molecular do polímero é utilizado um tipo de mercaptan como agente de transferência de cadeia. A reação de polimerização a frio de SBR é geralmente conduzida até conversões na faixa de 60% e então são adicionados os agentes de terminação que atacam os radicais livres e interrompem a propagação das cadeias. Dentre os principais agentes de terminação empregados podemos citar o dimetil-ditiocarbamato de sódio e o dietil-hidroxilamina (ROCHA; SOARES; COUTINHO, 2007). Abaixo segue uma tabela retirada da bibliografia com os valores clássicos das condições de processo e da formulação da alimentação da reação de polimerização de borracha SBR via emulsão a frio.

**Tabela A.1:** Condições de Processo e componentes utilizados na polimerização a frio de SBR

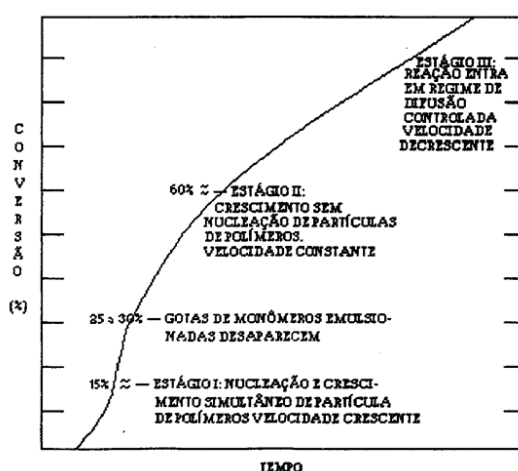
Condições e componentes do processo	SBR a frio
Temperatura de polimerização (°C)	5
Tempo de polimerização (h)	12
Conversão (%)	60-65
Butadieno (%)	71
Estireno (%)	29
Água (pcm)	190
Emulsificante (pcm)	4,5-5
Mercaptan de t-dodecila (pcm)	0,2
Hidroperóxido de p-mentila	0,08
Fosfato de sódio (pcm)	0,5
Sulfato Ferroso (pcm)	0,4
Formaldeído sulfoxilato de sódio	0,1
EDTA	0,06

Fonte: (ROCHA; SOARES; COUTINHO, 2007)

#### A.1.2 Etapas da polimerização em emulsão a frio via radicais livres

Para a polimerização de borracha SBR em emulsão o mecanismo mais aceito é o da nucleação micelar ou heterogênea (SMITH; EWART, 1948), uma teoria qualitativa resultante de observações experimentais de polimerização de estireno e de outros monômeros que possuem baixa solubilidade em água (HARKINS,1947). Este mecanismo de reação é predominante quando há condições de concentração micelar acima da concentração micelar crítica (CMC) e quando há entrada dos radicais livres formados no sistema de iniciação para o interior das micelas (HERRERA-ORDONEZ; OLAYO,2000,2001). Podem ser observados três estágios distintos no mecanismo de nucleação micelar (GARDON,1968).

Figura A.1: Etapas da Polimerização em emulsão via radicais livres



Fonte: (SMITH; EWART, 1948)

No primeiro estágio da reação, denominado Nucleação de Partículas, os radicais livres gerados na fase aquosa (ou contínua) migram para o interior das micelas de sabão ocorrendo assim o início da polimerização dos monômeros (1,3-butadieno e estireno). A partir deste ponto, as micelas se transformam em partículas de polímero estabilizadas pelo emulsificante. Nesta etapa, observa-se aumento da velocidade de polimerização com o tempo e ao seu final todas as micelas forma consumidas e a geração de novas partículas de polímero acaba, o que ocorre com conversão entre 10-20% no caso da copolimerização de 1,3-butadieno com estireno (SMITH; EWART, 1948).

Na etapa seguinte, denominada Crescimento das Partículas, ocorre o crescimento das partículas de polímero abastecido pelos reservatórios de monômeros, que mantém as partículas poliméricas saturadas de monômeros pelo mecanismo de difusão a partir da fase aquosa. Se consegue assim que a concentração de monômeros no interior das partículas de polímero se mantenha constante ao decorrer desta etapa, mantendo assim a velocidade de reação também constante. No final deste estágio, ocorre o desaparecimento das macrogotas de monômeros (reservatórios) e todos os monômeros que ainda não reagiram se encontram no interior das partículas poliméricas, o que ocorre com conversão de aproximadamente 50-60% para a copolimerização de 1,3-butadieno com estireno (SMITH; EWART, 1948).

Na última etapa, denominada Término da Polimerização, é observada a queda contínua da concentração de monômeros no interior das partículas poliméricas, o que faz com que a velocidade de reação também decresça. A conclusão desta etapa é controlada pela adição de terminador na mistura reacional quando a conversão de 60-65% é alcançada (SMITH; EWART, 1948).

### *A.1.3 Características controladas na reação de SBR*

Existem particularmente 3 (três) variáveis de interesse que são determinadas a partir de análises laboratoriais no processo de polimerização de borracha SBR: a viscosidade Mooney do elastômero, o teor de sólidos totais e o teor de estireno combinado na composição da borracha. Estas variáveis de interesse acima descritas são as quais o analisador virtual desenvolvido no presente trabalho pretende estimar a partir de um modelo matemático. A seguir são descritas cada uma destas características da borracha.

- **Teor de Sólidos Totais**

Na produção de borracha SBR o teor de sólidos totais é uma medida direta da conversão dos monômeros adicionados na reação. Este ensaio consiste na coleta de uma amostra de látex não-esgotado (látex pós-reação com a presença de monômeros não reagidos) a jusante do último reator da cadeia de reação em um vaso de pressão onde ele é evacuado até uma pressão negativa de aproximadamente -550 mmHg para a remoção do 1,3-butadieno não reagido da amostra. Este látex é então encaminhado ao laboratório onde é pesada a massa inicial em uma cápsula de alumínio e então introduzida em uma estufa com lâmpadas de secagem infravermelha. A amostra é submetida à secagem por determinado período de tempo onde os compostos voláteis são liberados para um duto de exaustão e permanece na cápsula apenas os compostos não-voláteis, majoritariamente a borracha SBR propriamente dita. A fórmula para o cálculo do teor de sólidos é a seguinte (WISNIEWSKI; ROHNELT, 1947):

$$ST\% = \frac{\text{massa final}(g) - \text{massa cápsula}(g)}{\text{massa inicial}(g) - \text{massa cápsula}(g)} \times 100\% \quad (\text{a.1})$$

Onde:  $ST\%$  = teor de sólidos totais no látex

Considerando a receita de polimerização de borracha SBR utilizado no processo objeto deste estudo, chegamos ao valor de sólidos totais na faixa de 22,0% a 23,0% para que seja obtida uma conversão dos monômeros no entorno de 60% a 66%. Caso o teor de sólidos totais se encontre fora da faixa de especificação, o operador eleva a vazão de reação (reduzindo assim o tempo de residência do látex dentro de cada reator) para reduzir o teor de sólidos ou reduz a vazão de reação (aumentando assim o tempo de residência do látex em cada reator) para aumentar o teor de sólidos totais.

A determinação do teor de sólidos totais é de especial importância para o desenvolvimento do analisador virtual proposto pois a conversão de monômeros está relacionada com a viscosidade Mooney do polímero. Quanto maior a conversão, maior será a viscosidade Mooney. Este fato ocorre, pois, a viscosidade Mooney está intimamente relacionada ao peso molecular do polímero e para sistemas de polimerização em cadeia quanto maior a conversão de monômeros, maior o peso molecular do polímero e consequentemente maior será a viscosidade Mooney.

- **Teor de Estireno Combinado (EC)**

O teor de estireno combinado na borracha SBR é obtido por refratometria (refratômetro Leica Abbe Mark II) conduzido a temperatura de 25°C, sendo que o estireno é adicionado em conjunto com o 1,3-butadieno na composição do copolímero para conferir características mecânicas de maior dureza, resistência à abrasão e processabilidade (MARTINS PALHA JÚNIOR, 2003). No processo aqui estudado, o teor de estireno combinado deve se encontrar na faixa de 22,5% a 23,5%. Caso o resultado encontrado na análise de refratometria indique um valor fora da faixa de especificação, o operador de processo altera o valor da chamada relação Butadieno-estireno, que nada mais é que os valores de pcm de Butadieno e estireno respectivamente. Caso haja necessidade de elevação do teor de estireno combinado (EC), eleva-se o pcm de estireno e se reduz o de Butadieno e caso haja necessidade de redução do EC, reduz-se o pcm de estireno e se eleva o de Butadieno. Lembrando que os valores de pcm de ambos os monômeros somados devem totalizar 100 (cem) partes de monômero.

## APÊNDICE B

### B.1 Código de Implementação da Metodologia em Python

```
### Importando pacotes úteis
import pandas as pd
import numpy as np
np.random.seed(42)
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('seaborn-whitegrid')
color = sns.color_palette()
import warnings
def ignore_warn(*args, **kwargs):
    pass
warnings.warn = ignore_warn #ignore annoying warning (from sklearn and seaborn)
from scipy import stats
from scipy.stats import norm, skew
from sklearn.ensemble import GradientBoostingRegressor
from xgboost import XGBRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.decomposition import PCA
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet, BayesianRidge
from sklearn.ensemble import GradientBoostingRegressor, AdaBoostRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn import metrics
from sklearn import preprocessing

from google.colab import drive
drive.mount('/content/drive')
```

Show hidden output

## ▼ Pré-tratamento dos dados e Seleção de Variáveis

```
df = pd.read_excel("/content/drive/MyDrive/TCC /Base_de_dados2.xlsx") # Grupo de dados de tre
val = pd.read_excel("/content/drive/MyDrive/TCC /Validation_new.xlsx") # Novo grupo de dados
```

```
### Removendo outliers vazões
df.drop(df.loc[df[0]<10].index, inplace=True)
df.drop(df.loc[df[0]>24].index, inplace=True)
df.describe()
```

```
### Novas variáveis para media das temperaturas
df[45]=(df[10]+df[11]+df[12]+df[13]+df[14])/5
df[46]=(df[15]+df[16]+df[17]+df[18]+df[19])/5
```

```
df[47]=(df[20]+df[21]+df[22]+df[23]+df[24])/5
df.describe()
```

Show hidden output

```
plt.figure(figsize=(8, 12))
y = df[0]
fig= plt.figure(figsize=(30,6))
plt.plot( y, '-', color='black')
```

Show hidden output

```
df.index=range(0,57186,1)
```

```
### Removendo outliers das temperaturas
```

```
i=0
```

```
while i in range (57186):
    if df[10][i]<8 or df[10][i]>14:
        df[10][i]= df[10].mean()
    if df[11][i]<8 or df[11][i]>14:
        df[11][i]= df[10][i]
    if df[12][i]<8 or df[12][i]>14:
        df[12][i]= df[11][i]
    if df[13][i]<8 or df[13][i]>14:
        df[13][i]= df[12][i]
    if df[14][i]<8 or df[14][i]>14:
        df[14][i]= df[13][i]
    if df[15][i]<8 or df[15][i]>14:
        df[15][i]= df[14][1]
    if df[16][i]<8 or df[16][i]>14:
        df[16][i]= df[15][i]
    if df[17][i]<8 or df[17][i]>14:
        df[17][i]= df[16][i]
    if df[18][i]<8 or df[18][i]>14:
        df[18][i]= df[17][i]
    if df[19][i]<8 or df[19][i]>14:
        df[19][i]= df[18][i]
    if df[20][i]<8 or df[20][i]>14:
        df[20][i]= df[19][i]
    if df[21][i]<8 or df[21][i]>14:
        df[21][i]= df[20][i]
    if df[22][i]<8 or df[22][i]>14:
        df[22][i]= df[21][i]
    if df[23][i]<8 or df[23][i]>14:
        df[23][i]= df[22][i]
    if df[24][i]<8 or df[24][i]>14:
        df[24][i]= df[23][i]
```



```

i=i+1

dfx=df
dfx=dfx.interpolate(method='linear')
dfx=dfx.fillna(method='backfill')
dfx=dfx.drop(['t'],axis=1)
#Plot da variável VM e ST
plt.figure(figsize=(8, 12))
y = dfx[42]
y2 = df[42]
fig= plt.figure(figsize=(30,6))
plt.plot( y, '-', color='black')
plt.plot( y2, 'o', color='red')

plt.figure(figsize=(8, 12))
y = dfx[43]
y2 = df[43]
fig= plt.figure(figsize=(30,6))
plt.plot( y, '-', color='black')
plt.plot( y2, 'o', color='red')

Show hidden output

# Escalonando dados (média=0 e desvpad =1)

scaler = preprocessing.StandardScaler().fit(dfx)
dfx = scaler.transform(dfx)
dfx=pd.DataFrame(dfx)

# Analisando deslocamentos de tempo ótimos
j=0
while j in range (25):
    corrMatrix = dfx.corr(method='pearson')
    i=0
    a=np.zeros(21)
    b=np.zeros(21)
    t=(0.0,0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,8.0,8.5,9.0,9.5,10.0)
    while i<21:
        a[i]=corrMatrix[42][j]
        b[i]=corrMatrix[43][j]
        dfx[j]=dfx[j].shift(6)
        corrMatrix = dfx.corr(method='pearson')
        i=i+1

dcorr=pd.DataFrame({'corrVM':a,'corrST':b,'lag':t})
y=abs(dcorr['corrVM'])
x=dcorr['lag']
plt.plot(x, y, 'black')
plt.plot(x, y, 'r')
plt.xlabel('lag time (h)')

```

```
plt.xlabel('lag time (h)')
plt.ylabel('correlation between %s and 42'% j)
plt.show()

y=abs(dcorr['corrST'])
x=dcorr['lag']
plt.plot(x, y, 'black')
plt.xlabel('lag time (h)')
plt.ylabel('correlation between %s and 43'% j)
plt.show()
j=j+1
```

Show hidden output

```
#Deslocamentos de tempo ótimos
```

```
df[0]=df[0].shift(84)
df[1]=df[1].shift(84)
df[2]=df[2].shift(84)
df[3]=df[3].shift(84)
df[4]=df[4].shift(84)
df[5]=df[5].shift(84)
df[7]=df[7].shift(84)
df[8]=df[8].shift(84)
df[9]=df[9].shift(84)
df[10]=df[10].shift(72)
df[11]=df[11].shift(60)
df[12]=df[12].shift(72)
df[13]=df[13].shift(66)
df[14]=df[14].shift(60)
df[15]=df[15].shift(48)
df[16]=df[16].shift(60)
df[17]=df[17].shift(60)
df[18]=df[18].shift(48)
df[19]=df[19].shift(54)
df[20]=df[20].shift(48)
df[21]=df[21].shift(36)
df[22]=df[22].shift(30)
df[23]=df[23].shift(30)
df[24]=df[24].shift(24)
df[40]=df[40].shift(60)
# Inverso do quadrado da temperatura para tentar obter o comportamento não linear das tempera
df[48]=1/((df[17])**2)
df[49]=1/((df[19])**2)
df[50]=1/((df[23])**2)
df[51]=1/((df[24])**2)
```

```
# Criando dataset para valores de laboratório (a serem usados para adaptação do modelo)
lab=df[42]
#lab = lab.fillna(0)
df1=df.interpolate(method='linear')
```

```
df1=df1.fillna(method='backfill')
df[0]=df[0].fillna(method='backfill')
df1=df1.drop('t',axis=1)

# Variável para simular concentração de modificador
i=0
j=0
k=0
df[52]=df[0]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[0][k]
            df[52][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de butadieno
i=0
j=0
k=0
df[55]=df[2]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[2][k]
            df[55][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de emulsificante
i=0
j=0
k=0
df[56]=df[4]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[4][k]
            df[56][i] = (sum(s))/50
```

```
j=j+1
i=i+1

# Variável para simular concentração de ativador
i=0
j=0
k=0
df[57]=df[5]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[5][k]
            df[57][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de água
i=0
j=0
k=0
df[58]=df[7]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[7][k]
            df[58][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular vazão média de amônia vaporizada
i=0
j=0
k=0
df[59]=df[9]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[9][k]
            df[59][i] = (sum(s))/50
```

```
j=j+1
i=i+1
```

```
# Variável para simular temperatura média Reator 7
```

```
i=0
j=0
k=0
df[60]=df[17]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[17][k]
            df[60][i] = (sum(s))/50
            j=j+1
        i=i+1
```

```
# Variável para simular temperatura média Reator 9
```

```
i=0
j=0
k=0
df[61]=df[19]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[19][k]
            df[61][i] = (sum(s))/50
            j=j+1
        i=i+1
```

```
# Variável para simular temperatura média Reator 13
```

```
i=0
j=0
k=0
df[62]=df[23]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[23][k]
            df[62][i] = (sum(s))/50
```

```

    j=j+1
    i=i+1

# Variável para simular temperatura média Reator 14
i=0
j=0
k=0
df[63]=df[24]
s = np.zeros(50)
while i in range(57186):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = df[24][k]
            df[63][i] = (sum(s))/50
            j=j+1
        i=i+1

#Standardizing data (CONJUNTO COMPLETO)
from sklearn import preprocessing
scaler = preprocessing.StandardScaler().fit(df1.drop(42,axis=1))
ds = scaler.transform(df1.drop(42,axis=1))
ds=pd.DataFrame(ds)
ds.columns=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30]
ds[42]=df1[42]

ds.describe()

Show hidden output

from sklearn.decomposition import PCA
pca = PCA(n_components=15)
pca.fit(ds)
x_pca = pca.transform(ds)
##### PLOTAR PCA

def _plot_pca(pca):
    plt.figure(figsize=(25, 10), dpi=80, facecolor='w', edgecolor='k')
    plt.plot(range(pca.n_components_),np.cumsum(pca.explained_variance_ratio_), '-o', label =
    plt.bar(range(pca.n_components_),pca.explained_variance_ratio_,width = 0.8, align = 'cent
    #plt.bar(range(pca.n_components_),np.cumsum(pca.explained_variance_ratio_)-pca.explained_
    plt.tick_params(axis='x', which='major', labelsize=8)
    #plt.plot(range(pca.n_components_),np.ones([pca.n_components_])*0.95, c = 'r')
    plt.plot([-0.4,pca.n_components_+1],[0.99, 0.99], c = 'r', ls = '--', label = '99% de var
    plt.xlim([-0.4,pca.n_components_+1])
    leg = plt.legend(loc='center right', fontsize = 12)

```

```
leg.get_frame().set_alpha(0.5)
plt.xlabel('Número de Componentes Principais', size = 13)
plt.ylabel('Variância Explicada ( % )', size = 13)
```

```
pcs = PCA()
pcs.fit_transform(ds)
_plot_pca(pcs)
```

Show hidden output

### PCA das temperaturas

```
T = pd.DataFrame(df1, columns=[10,11,12,13,14,15,16,17,18,19,20,21,22,23,24])
scaler = preprocessing.StandardScaler().fit(T)
Ts = scaler.transform(T)
Ts=pd.DataFrame(Ts)
```

```
def _plot_pca(pca):
    plt.figure(figsize=(13, 5), dpi=80, facecolor='w', edgecolor='k')
    plt.plot(range(pca.n_components_),np.cumsum(pca.explained_variance_ratio_), '-o', label =
    plt.bar(range(pca.n_components_),pca.explained_variance_ratio_,width = 0.8, align = 'cent
    #plt.bar(range(pca.n_components_),np.cumsum(pca.explained_variance_ratio_)-pca.explained_
    plt.tick_params(axis='x', which='major', labelsize=8)
    #plt.plot(range(pca.n_components_),np.ones([pca.n_components_] *0.95, c = 'r')
    plt.plot([-0.4,pca.n_components_+1],[0.99, 0.99], c = 'r', ls = '--', label = '99% de var
    plt.xlim([-0.4,pca.n_components_+1])
    leg = plt.legend(loc='center right', fontsize = 12)
    leg.get_frame().set_alpha(0.5)
    plt.xlabel('Número de Componentes Principais', size = 13)
    plt.ylabel('Variância Explicada ( % )', size = 13)
```

```
pcs = PCA()
pcs.fit_transform(Ts)
_plot_pca(pcs)
```

Show hidden output

```
df2=df
df2=df2.drop('t',axis=1)
df2=df2.drop(44,axis=1)
df2.dropna(how='any',inplace=True)
df2.describe()
```

Show hidden output

```
scaler = preprocessing.StandardScaler().fit(df2.drop(42,axis=1))
ds2 = scaler.transform(df2.drop(42,axis=1))
ds2=pd.DataFrame(ds2)
ds2.columns=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,
ds2.index=df2.index
```

```
ds2[42]=df2[42]
ds2[54]=np.log(df2[42])
ds2.describe()
```

Show hidden output

```
from sklearn import linear_model
from sklearn.linear_model import Lasso
i=0.000001
j=0
k=0
a=np.zeros(7)
r=np.zeros(7)
n=np.zeros(7) #número de variáveis diferentes de zero
X = ds.drop([42],axis=1).values
y=ds[42].values
```

```
while i<=1.0:
    reg=linear_model.LassoLars(alpha=i)
    reg.fit(X,y)
    a[j]=i
    r[j]=reg.score(X,y)
    C=reg.coef_
    for k in range(42):
        if C[k]!=0:
            n[j]=n[j]+1
    i=i*10
    j=j+1
```

```
ev=pd.DataFrame({'alpha':a, 'R2':r})
```

```
y=ev['R2']
x=np.log10(ev['alpha'])
plt.plot(x, y, 'black')
plt.xlabel('log(alpha)')
plt.ylabel('R²')
plt.show()
```

```
plt.bar(x,n)
plt.xlabel('log(alpha)')
plt.ylabel('n de var dif. de zero')
plt.show()
```

Show hidden output

```
i=1e-3
j=0
k=0
a=np.zeros(10)
r=np.zeros(10)
```



```
n=np.zeros(10) #número de variáveis diferentes de zero
X = ds.drop([42],axis=1).values
y=ds[42].values
```

```
while i<=2e-3:
    reg=linear_model.LassoLars(alpha=i)
    reg.fit(X,y)
    a[j]=i
    r[j]=reg.score(X,y)
    C=reg.coef_
    for k in range(42):
        if C[k]!=0:
            n[j]=n[j]+1
    i=i+1e-4
    j=j+1
```

```
ev2=pd.DataFrame({'alpha':a,'R2':r,'var_dif_0':n})
y=ev2['R2']
x=ev2['alpha']
plt.plot(x, y, 'black')
plt.xlabel('alpha')
plt.ylabel('R²')
plt.show()
```

ev2

Show hidden output

```
X=ds.drop([42],axis=1).values
y=ds[42].values
reg=linear_model.LassoLars(alpha=0.0015)
reg.fit(X,y)
VM=reg.coef_
i=0
j=0
var=np.zeros(14)
while i<42:
    if VM[i]!=0:
        var[j]=i
        j=j+1
    i=i+1
var
```

Show hidden output

```
VM = pd.DataFrame(ds2,columns=[0,2,4,5,7,9,17,19,23,24,25,30,38,39,42])
VM.describe()
```

Show hidden output

VM.index

Show hidden output

```
corrMatrix1 = ds.corr(method='pearson')
plt.figure(figsize=(8, 12))
heatmap = sns.heatmap(corrMatrix1[[42]].sort_values(by=42, ascending=False), vmin=-1, vmax=1,
heatmap.set_title('Features Correlating with Mooney Viscosity (df)', fontdict={'fontsize':18})
plt.figure(figsize=(8, 12))
```

Show hidden output

```
df[52].describe()
```

Show hidden output

```
df[0].describe()
```

Show hidden output

```
## Avaliar a distribuição das variáveis selecionadas (quanto mais normal melhor)
## poderia aplicar transformações para deixar linear (ex. log) - mas vamos deixar assim
pd.DataFrame(VM,columns=VM.columns).hist(figsize=(15,10));
```

Show hidden output

```
VM.describe()
```

Show hidden output

## ▼ Elaboração do Modelo Variável 42 - Viscosidade Mooney

```
X = VM.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]
train.shape, test.shape
```

Show hidden output

```
df_train_div = pd.DataFrame(train, columns=VM.columns, index=VM.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=VM.columns, index=VM.index[train.shape[0]:])
```

```
## para 42
ytest = df_test_div[42].values
ytrain = df_train_div[42].values
```

```
xtrain = df_train_div.drop([42], axis=1)

lr = LinearRegression().fit(xtrain,ytrain)

ypred1 = lr.predict(xtest)
#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred1
results.drop(results.loc[results[1]<40].index, inplace=True)
results.drop(results.loc[results[1]>70].index, inplace=True)
results.describe()
```

```
ytest01 = results[0].values
ypred1=results[1].values
r2_score(ytest,ypred1)
```

Show hidden output

```
mean_squared_error(ytest,ypred1)
```

Show hidden output

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(ytest,ypred1)
```

Show hidden output

```
from sklearn.metrics import explained_variance_score
explained_variance_score(ytest,ypred1)
```

Show hidden output

```
VM = pd.DataFrame(ds2,columns=[0,2,4,5,7,11,17,25,30,38,39,40,42])
X = VM.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]
train.shape, test.shape
```

```
df_train_div = pd.DataFrame(train, columns=VM.columns, index=VM.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=VM.columns, index=VM.index[train.shape[0]:])
```

```
## para 42
ytest = df_test_div[42].values
ytrain = df_train_div[42].values
xtrain = df_train_div.drop([42], axis=1)
xtest = df_test_div.drop([42], axis=1)
```

```
gb = GradientBoostingRegressor(n_estimators=69, learning_rate=0.1, loss='ls',random_state=42)
```

```
ypred2 = gb.predict(xtest)
r2_score(ytest,ypred2)
```

Show hidden output

```
mean_squared_error(ytest,ypred2)
```

Show hidden output

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(ytest,ypred2)
```

Show hidden output

```
from sklearn.metrics import explained_variance_score
explained_variance_score(ytest,ypred2)
```

Show hidden output

```
xgb = XGBRegressor(max_depth=3, learning_rate=0.11, n_estimators=117, booster='gbtree', n_job
ypred3 = xgb.predict(xtest)
r2_score(ytest,ypred3)
```

Show hidden output

```
mean_squared_error(ytest,ypred3)
```

Show hidden output

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(ytest,ypred3)
```

Show hidden output

```
from sklearn.metrics import explained_variance_score
explained_variance_score(ytest,ypred3)
```

Show hidden output

```
plt.figure(figsize=(30, 10), dpi=80, facecolor='w', edgecolor='k')
plt.style.use('seaborn-whitegrid')
plt.plot(ytest, 'o--', label = 'Dados',linewidth=0.5, c='black')
plt.plot(ypred2,'s', label = 'GradientBoosting',linewidth=2.0,c = 'green')
plt.plot(ypred3,'p', label = 'XGB',linewidth=2.0,c = 'blue')
plt.plot(ypred1,'^', label = 'MLR',linewidth=2.0,c = 'red')
plt.tick_params(axis='x', which='major', labelsize=15)
plt.tick_params(axis='y', which='major', labelsize=15)
leg = plt.legend(loc='upper right', fontsize = 17)
leg.get_frame().set_alpha(0.5)
plt.xlabel('Número da Amostra', size = 18)
```

```
plt.ylabel('Viscosidade Mooney', size = 18)
plt.axis([0,350,35,75])

plt.figure(figsize=(15, 10), dpi=80, facecolor='w', edgecolor='k')
plt.scatter(ypred1, ytest01, c = "black")
plt.xlabel("Predicted values")
plt.ylabel("Real values")
plt.plot([40,80], [40, 80], c = "red")
plt.show()
```

Show hidden output

```
VM = pd.DataFrame(ds2,columns=[0,2,4,5,7,9,17,23,24,25,30,38,39,42,45,46,47])### Inserindo a
```

```
X = VM.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]
train.shape, test.shape

df_train_div = pd.DataFrame(train, columns=VM.columns, index=VM.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=VM.columns, index=VM.index[train.shape[0]:])
```

```
## para 42
ytest = df_test_div[42].values
ytrain = df_train_div[42].values
xtrain = df_train_div.drop([42], axis=1)
xtest = df_test_div.drop([42], axis=1)
```

```
lr = LinearRegression().fit(xtrain,ytrain)
ypred5 = lr.predict(xtest)
#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred5
results.drop(results.loc[results[1]<40].index, inplace=True)
results.drop(results.loc[results[1]>70].index, inplace=True)
results.index = range(0,336,1)
results.describe()
```

```
ytest1 = results[0].values
ypred5=results[1].values
r2_score(ytest1,ypred5)
```

Show hidden output

```
mean_squared_error(ytest1,ypred5)
```

Show hidden output

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(ytest1,ypred5)
```

Show hidden output

```
from sklearn.metrics import explained_variance_score
explained_variance_score(ytest1,ypred5)
```

Show hidden output

```
VM = pd.DataFrame(ds2,columns=[0,2,4,5,7,9,17,23,24,25,30,38,39,42,45,46,47,48,49,50,51])###
```

```
X = VM.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]
train.shape, test.shape
```

```
df_train_div = pd.DataFrame(train, columns=VM.columns, index=VM.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=VM.columns, index=VM.index[train.shape[0]:])
```

```
## para 42
ytest = df_test_div[42].values
ytrain = df_train_div[42].values
xtrain = df_train_div.drop([42], axis=1)
xtest = df_test_div.drop([42], axis=1)
```

```
lr = LinearRegression().fit(xtrain,ytrain)
ypred6 = lr.predict(xtest)
```

```
#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred6
results.drop(results.loc[results[1]<40].index, inplace=True)
results.drop(results.loc[results[1]>70].index, inplace=True)
results.index = range(0,336,1)
results.describe()
```

```
ytest2 = results[0].values
ypred6=results[1].values
```

```
r2_score(ytest2,ypred6)
```

Show hidden output

```
mean_squared_error(ytest2,ypred6)
```

Show hidden output

```
from sklearn.metrics import mean_absolute_error
```

```
mean_absolute_error(ytest2,ypred6)
```

Show hidden output

```
from sklearn.metrics import explained_variance_score
explained_variance_score(ytest2,ypred6)
```

Show hidden output

```
VM = pd.DataFrame(ds2,columns=[0,2,4,5,7,9,17,19,23,24,25,30,38,39,42,45,46,47,52])### Inseri
```

```
X = VM.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]
train.shape, test.shape
```

```
df_train_div = pd.DataFrame(train, columns=VM.columns, index=VM.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=VM.columns, index=VM.index[train.shape[0]:])
```

```
## para 42
ytest = df_test_div[42].values
ytrain = df_train_div[42].values
xtrain = df_train_div.drop([42], axis=1)
xtest = df_test_div.drop([42], axis=1)
```

```
lr = LinearRegression().fit(xtrain,ytrain)
ypred10 = lr.predict(xtest)
```

```
#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred10
results.drop(results.loc[results[1]<40].index, inplace=True)
results.drop(results.loc[results[1]>70].index, inplace=True)
results.index = range(0,336,1)
results.describe()
```

```
ytest10 = results[0].values
ypred10=results[1].values
```

```
r2_score(ytest10,ypred10)
```

Show hidden output

```
mean_squared_error(ytest10,ypred10)
```

Show hidden output

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(ytest10,ypred10)
```

Show hidden output

```
from sklearn.metrics import explained_variance_score
explained_variance_score(ytest10,ypred10)
```

Show hidden output

```
plt.figure(figsize=(30, 10), dpi=80, facecolor='w', edgecolor='k')
plt.style.use('seaborn-whitegrid')
plt.plot(ytest10, '-o', label = 'Dados',linewidth=1.0, c='black')
plt.plot(ypred10, '--o', label = 'Conjunto de dados com var. 52',linewidth=1.5,c = 'green')
#plt.plot(ypred1, '--o', label = 'Conjunto de dados sem var. 52',linewidth=1.5,c = 'magenta')
plt.tick_params(axis='x', which='major', labelsize=15)
plt.tick_params(axis='y', which='major', labelsize=15)
leg = plt.legend(loc='upper right', fontsize = 17)
leg.get_frame().set_alpha(0.5)
plt.xlabel('Número da Amostra', size = 18)
plt.ylabel('Viscosidade Mooney', size = 18)
plt.axis([0,232,40,73])
```

```
plt.figure(figsize=(15, 10), dpi=80, facecolor='w', edgecolor='k')
plt.scatter(ypred10, ytest10, c = "black")
plt.xlabel("Predicted values")
plt.ylabel("Real values")
plt.plot([40,70], [40, 70], c = "red")
plt.show()
```

Show hidden output

```
VM = pd.DataFrame(ds2,columns=[0,2,4,5,7,9,17,23,24,25,30,38,39,45,46,47,52,54])### Predição
```

```
X = VM.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]
train.shape, test.shape
```

```
df_train_div = pd.DataFrame(train, columns=VM.columns, index=VM.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=VM.columns, index=VM.index[train.shape[0]:])
```

```
## para 42
ytest = df_test_div[54].values
ytrain = df_train_div[54].values
xtrain = df_train_div.drop([54], axis=1)
xtest = df_test_div.drop([54], axis=1)
```

```
lr = LinearRegression().fit(xtrain,ytrain)
ypred11 = lr.predict(xtest)
```



```

ypred11 = np.exp(ypred11)

#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred11
results.drop(results.loc[results[1]<40].index, inplace=True)
results.drop(results.loc[results[1]>70].index, inplace=True)
results.index = range(0,336,1)
results.describe()

```

```

ytest10 = results[0].values
ypred11=results[1].values

```

```
r2_score(ytest2,ypred11)
```

Show hidden output

```

plt.figure(figsize=(30, 10), dpi=80, facecolor='w', edgecolor='k')
plt.style.use('seaborn-whitegrid')
plt.plot(ytest2, '-o', label = 'Dados',linewidth=1.0, c='black')
plt.plot(ypred1, '--o', label = 'MLR convencional',linewidth=1.5,c = 'blue')
plt.plot(ypred11, '--o', label = 'modelo var. LN',linewidth=1.5,c = 'red')
plt.tick_params(axis='x', which='major', labelsize=15)
plt.tick_params(axis='y', which='major', labelsize=15)
leg = plt.legend(loc='upper right', fontsize = 17)
leg.get_frame().set_alpha(0.5)
plt.xlabel('Número da Amostra', size = 18)
plt.ylabel('Viscosidade Mooney', size = 18)
plt.axis([0,232,40,73])

```

```

plt.figure(figsize=(15, 10), dpi=80, facecolor='w', edgecolor='k')
plt.scatter(ypred11, ytest2, c = "black")
plt.xlabel("Predicted values")
plt.ylabel("Real values")
plt.plot([40,70], [40, 70], c = "red")
plt.show()

```

Show hidden output

```
VM = pd.DataFrame(ds2,columns=[0,23,34,25,30,38,39,42,45,46,47,52,55,56,57,58,59,60,61,62,63])
```

```

X = VM.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]
train.shape, test.shape

```

```

df_train_div = pd.DataFrame(train, columns=VM.columns, index=VM.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=VM.columns, index=VM.index[train.shape[0]:])

```

```
## para 42
ytest = df_test_div[42].values
ytrain = df_train_div[42].values
xtrain = df_train_div.drop([42], axis=1)
xtest = df_test_div.drop([42], axis=1)

lr = LinearRegression().fit(xtrain,ytrain)
ypred12 = lr.predict(xtest)

#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred12
results.drop(results.loc[results[1]<40].index, inplace=True)
results.drop(results.loc[results[1]>70].index, inplace=True)
results.index = range(0,336,1)
results.describe()

ytest12 = results[0].values
ypred12=results[1].values

r2_score(ytest12,ypred12)

Show hidden output

mean_absolute_error(ytest12,ypred12)

Show hidden output

mean_squared_error(ytest12,ypred12)

Show hidden output

explained_variance_score(ytest12,ypred12)

Show hidden output

plt.figure(figsize=(30, 10), dpi=80, facecolor='w', edgecolor='k')
plt.style.use('seaborn-whitegrid')
plt.plot(ytest12, 'o', label = 'Dados',linewidth=0.5, c='black')
plt.plot(ypred12, 's', label = 'MLR com melhor conjunto de dados',linewidth=1.5,c = 'magenta')
#plt.plot(ypred1, '--o', label = 'Conjunto de dados sem var. 52',linewidth=1.5,c = 'magenta')
plt.tick_params(axis='x', which='major', labelsize=15)
plt.tick_params(axis='y', which='major', labelsize=15)
leg = plt.legend(loc='upper right', fontsize = 17)
leg.get_frame().set_alpha(0.5)
plt.xlabel('Número da Amostra', size = 18)
plt.ylabel('Viscosidade Mooney', size = 18)
plt.axis([0,350,40,73])
```

```
plt.figure(figsize=(15, 10), dpi=80, facecolor='w', edgecolor='k')
plt.scatter(ypred12, ytest12, c = "black")
plt.xlabel("Predicted values")
plt.ylabel("Real values")
plt.plot([40,70], [40, 70], c = "red")
plt.show()
```

Show hidden output

## ▼ Modelo para Inferência dos Resíduos

```
residuo = ytest12 - ypred12
res=pd.DataFrame(residuo)
res.describe()
```

Show hidden output

```
X = ds2.values
R = X[822:1159]
R = pd.DataFrame(R, columns = ds2.columns)
R[53]= res
```

```
R.describe()
```

Show hidden output

```
# Seleção de variáveis para o modelo dos resíduos
from sklearn import linear_model
from sklearn.linear_model import Lasso
i=0.000001
j=0
k=0
a=np.zeros(7)
r=np.zeros(7)
n=np.zeros(7) #número de variáveis diferentes de zero
X = R.drop([42,53],axis=1).values
y=R[53].values
```

```
while i<=1.0:
    reg=linear_model.LassoLars(alpha=i)
    reg.fit(X,y)
    a[j]=i
    r[j]=reg.score(X,y)
    C=reg.coef_
    for k in range(51):
        if C[k]!=0:
            n[j]=n[j]+1
```

```

i=i*10
j=j+1

ev=pd.DataFrame({'alpha':a,'R2':r})

y=ev['R2']
x=np.log10(ev['alpha'])
plt.plot(x, y, 'black')
plt.xlabel('log(alpha)')
plt.ylabel('R²')
plt.show()

plt.bar(x,n)
plt.xlabel('log(alpha)')
plt.ylabel('n de var dif. de zero')
plt.show()

```

Show hidden output

```

X=R.drop([42,53],axis=1).values
y=R[53].values
reg=linear_model.LassoLars(alpha=0.02)
reg.fit(X,y)
Cf=reg.coef_
i=0
j=0
var=np.zeros(3)
while i<51:
    if Cf[i]!=0:
        var[j]=i
        j=j+1
    i=i+1
var

```

Show hidden output

```

R= pd.DataFrame(R,columns=[30,43,51,53])

# train and test for residuals model
X = R.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]

df_train_div = pd.DataFrame(train, columns=R.columns, index=R.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=R.columns, index=R.index[train.shape[0]:])

ytest = df_test_div[53].values
y = df_train_div[53].values

```

```
X = df_train_div.drop([53], axis=1).values
xtest = df_test_div.drop([53], axis=1).values

res = LinearRegression().fit(X,y)
```

```
yr_pred= res.predict(xtest)
r2_score(ytest,yr_pred)
```

Show hidden output

```
# Plot outputs
plt.style.use('seaborn-whitegrid')
figsize=(30,7)
fig = plt.figure(figsize=figsize)
plt.plot(ytest, 'o',color='black', label='Dados')
plt.plot(yr_pred,'s',color='red', label='Predição dos resíduos')
plt.xlabel('número da amostra', size = 20)
plt.ylabel('Resíduo', size = 20)
plt.axis([0,100,-17,17])
leg = plt.legend(loc='upper right', fontsize = 17)
plt.tick_params(axis='x', which='major', labelsize=15)
plt.tick_params(axis='y', which='major', labelsize=15)
plt.show()
```

Show hidden output

```
VM = pd.DataFrame(ds2,columns=[0,23,24,25,30,38,39,42,45,46,47,52,55,56,57,58,59,60,61,62,63])
```

```
X = VM.values
train_size = int(len(X) * 0.71)
train, test = X[0:train_size], X[train_size:len(X)]
train.shape, test.shape
```

```
df_train_div = pd.DataFrame(train, columns=VM.columns, index=VM.index[:train.shape[0]])
df_test_div = pd.DataFrame(test, columns=VM.columns, index=VM.index[train.shape[0]:])
```

```
## para 42
ytest = df_test_div[42].values
ytrain = df_train_div[42].values
xtrain = df_train_div.drop([42], axis=1)
xtest = df_test_div.drop([42], axis=1)
```

```
lr = LinearRegression().fit(xtrain,ytrain)
ypred = lr.predict(xtest)
```

```
#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred
results.drop(results.loc[results[1]<40].index, inplace=True)
```

```
results.drop(results.loc[results[1]>70].index, inplace=True)
results.index = range(0,336,1)
results.describe()
```

```
r2_score(ytest,ypred)
```

Show hidden output

```
# Cálculo dos resíduos a partir do modelo de predição
xr = R.drop(53,axis=1)
yr_pred = res.predict(xr)
```

```
# Correção dos resultados apenas com a predição do modelo dos resíduos
ypred=ypred+yr_pred
```

```
#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred
results.drop(results.loc[results[1]<40].index, inplace=True)
results.drop(results.loc[results[1]>70].index, inplace=True)
results.index = range(0,334,1)
results.describe()
```

```
ytest100 = results[0].values
ypred100=results[1].values
```

```
r2_score(ytest,ypred)
```

Show hidden output

```
mean_squared_error(ytest,ypred)
```

Show hidden output

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(ytest,ypred)
```

Show hidden output

```
from sklearn.metrics import explained_variance_score
explained_variance_score(ytest,ypred)
```

Show hidden output

```
plt.figure(figsize=(25, 10), dpi=80, facecolor='w', edgecolor='k')
plt.style.use('seaborn-whitegrid')
plt.plot(ytest, 'o', label = 'Dados',linewidth=1.0, c='black')
```

```
plt.plot(ypred,'s', label = 'Caso B',linewidth=1.5,c = 'red')
plt.tick_params(axis='x', which='major', labelsize=8)
leg = plt.legend(loc='upper right', fontsize = 12)
leg.get_frame().set_alpha(0.5)
plt.xlabel('Número da Amostra', size = 13)
plt.ylabel('Viscosidade Mooney', size = 13)
plt.axis([0,350,35,75])

plt.figure(figsize=(15, 10), dpi=80, facecolor='w', edgecolor='k')
plt.scatter(ypred, ytest, c = "black")
plt.xlabel("Predicted values")
plt.ylabel("Real values")
plt.plot([40,70], [40, 70], c = "red")
plt.show()
```

Show hidden output

```
# Implementando correção de bias
```

```
y = ds2[42].values
lab = y[822:1159]
lab = pd.DataFrame(lab)
lab = lab.shift(1)
lab=lab.dropna()
lab.index=range(335)

i=0
bias=ytest[0]-ypred[0]

while i in range(335):
    if lab[0][i]!=0:
        bias=(lab[0][i]-ypred[i])*0.3 + bias*0.7
    else:
        bias=bias
    ypred[i]=ypred[i]+bias
    i=i+1

ypred7 = ypred

#remoção de outliers
results = pd.DataFrame(ytest)
results[1]=ypred7
results.drop(results.loc[results[1]<40].index, inplace=True)
results.drop(results.loc[results[1]>70].index, inplace=True)
results.index = range(0,336,1)
results.describe()

ytest4 = results[0].values
ypred7=results[1].values
ytest4=pd.DataFrame(ytest4)
```

```
ypred7=pd.DataFrame(ypred7)
ytest4=ytest4.dropna()
ypred7=ypred7.dropna()
```

```
ypred7.describe()
```

```
#r2_score(ytest4,ypred7)
```

Show hidden output

```
mean_squared_error(ytest4,ypred7)
```

Show hidden output

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(ytest4,ypred7)
```

Show hidden output

```
from sklearn.metrics import explained_variance_score
explained_variance_score(ytest4,ypred7)
```

Show hidden output

```
plt.figure(figsize=(25, 10), dpi=80, facecolor='w', edgecolor='k')
plt.style.use('seaborn-whitegrid')
plt.plot(ytest4, '-o', label = 'Dados',linewidth=1.0, c='black')
plt.plot(ypred7, '--o', label = 'Caso C',linewidth=1.5,c = 'magenta')
plt.tick_params(axis='x', which='major', labelsize=8)
leg = plt.legend(loc='upper right', fontsize = 17)
leg.get_frame().set_alpha(0.5)
plt.xlabel('Número da Amostra', size = 20)
plt.ylabel('Viscosidade Mooney', size = 20)
plt.tick_params(axis='x', which='major', labelsize=15)
plt.tick_params(axis='y', which='major', labelsize=15)
plt.axis([0,337,40.5,70])
```

```
plt.figure(figsize=(15, 10), dpi=80, facecolor='w', edgecolor='k')
plt.scatter(ypred7, ytest4, c = "black")
plt.xlabel("Predicted values")
plt.ylabel("Real values")
plt.plot([40,70], [40, 70], c = "red")
plt.show()
```

Show hidden output



# Aplicação da Metodologia para um novo grupo de dados (Validação)

```
val = pd.read_excel("/content/drive/MyDrive/TCC /Validation_new.xlsx") #Planilha completa

#Remoção dos outliers de temperaturas

i=0

while i in range (7875):
    if val[10][i]<8 or val[10][i]>14:
        val[10][i]= val[10].mean()
    if val[11][i]<8 or val[11][i]>14:
        val[11][i]= val[10][i]
    if val[12][i]<8 or val[12][i]>14:
        val[12][i]= val[11][i]
    if val[13][i]<8 or val[13][i]>14:
        val[13][i]= val[12][i]
    if val[14][i]<8 or val[14][i]>14:
        val[14][i]= val[13][i]
    if val[15][i]<8 or val[15][i]>14:
        val[15][i]= val[14][1]
    if val[16][i]<8 or val[16][i]>14:
        val[16][i]= val[15][i]
    if val[17][i]<8 or val[17][i]>14:
        val[17][i]= val[16][i]
    if val[18][i]<8 or val[18][i]>14:
        val[18][i]= val[17][i]
    if val[19][i]<8 or val[19][i]>14:
        val[19][i]= val[18][i]
    if val[20][i]<8 or val[20][i]>14:
        val[20][i]= val[19][i]
    if val[21][i]<8 or val[21][i]>14:
        val[21][i]= val[20][i]
    if val[22][i]<8 or val[22][i]>14:
        val[22][i]= val[21][i]
    if val[23][i]<8 or val[23][i]>14:
        val[23][i]= val[22][i]
    if val[24][i]<8 or val[24][i]>14:
        val[24][i]= val[23][i]
    i=i+1

val[0]=val[0].shift(84)
val[2]=val[2].shift(84)
val[4]=val[4].shift(84)
```

```
val[5]=val[5].shift(84)
val[7]=val[7].shift(84)
val[9]=val[9].shift(84)
val[10]=val[10].shift(72)
val[11]=val[11].shift(60)
val[12]=val[12].shift(72)
val[13]=val[13].shift(66)
val[14]=val[14].shift(60)
val[15]=val[15].shift(48)
val[16]=val[16].shift(60)
val[17]=val[17].shift(60)
val[18]=val[18].shift(48)
val[19]=val[19].shift(54)
val[20]=val[20].shift(48)
val[21]=val[21].shift(36)
val[22]=val[22].shift(30)
val[23]=val[23].shift(30)
val[24]=val[24].shift(24)
val[40]=val[40].shift(60)
### Novas variáveis para tentar detectar comportamento não linear das temperaturas
val[45]=(val[10]+val[11]+val[12]+val[13]+val[14])/5
val[46]=(val[15]+val[16]+val[17]+val[18]+val[19])/5
val[47]=(val[20]+val[21]+val[22]+val[23]+val[24])/5
## Deslocamento de tempo para aguardar execução da análise de laboratório
val[64] = val[42]
val[64] = val[64].shift(12)
```

```
val.describe()
```

Show hidden output

```
# Variável para simular concentração de modificador
i=0
j=0
k=0
val[52]=val[0]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[0][k]
            val[52][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de modificador
i=0
j=0
```

```
k=0
val[55]=val[2]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[2][k]
            val[55][i] = (sum(s))/50
            j=j+1
        i=i+1
```

```
# Variável para simular concentração de modificador
i=0
j=0
k=0
val[56]=val[4]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[4][k]
            val[56][i] = (sum(s))/50
            j=j+1
        i=i+1
```

```
# Variável para simular concentração de modificador
i=0
j=0
k=0
val[57]=val[5]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[5][k]
            val[57][i] = (sum(s))/50
            j=j+1
        i=i+1
```

```
# Variável para simular concentração de modificador
i=0
j=0
k=0
```

```
val[58]=val[7]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[7][k]
            val[58][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de modificador
i=0
j=0
k=0
val[59]=val[9]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[9][k]
            val[59][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de modificador
i=0
j=0
k=0
val[60]=val[17]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[17][k]
            val[60][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de modificador
i=0
j=0
k=0
val[61]=val[19]
```

```
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[19][k]
            val[61][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de modificador
i=0
j=0
k=0
val[62]=val[23]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[23][k]
            val[62][i] = (sum(s))/50
            j=j+1
        i=i+1

# Variável para simular concentração de modificador
i=0
j=0
k=0
val[63]=val[24]
s = np.zeros(50)
while i in range(7791):
    if i>50:
        j=0
        while j<50:
            k = i-j
            s[j] = val[24][k]
            val[63][i] = (sum(s))/50
            j=j+1
        i=i+1

val[51]=1/(val[24]**2)
val.drop(val[0].loc[val[0]<10].index, inplace=True)
val.drop(val[0].loc[val[0]>24].index, inplace=True)
val[64]=val[64].fillna(0)
labb = val[64]
```

```
ytest = val[42]
```

```
val=val.interpolate(method='linear')  
val=val.fillna(method='backfill')  
val=val.drop('t',axis=1)
```

```
val=pd.DataFrame(val,columns=[0,2,4,5,7,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,30,  
val.index=range(0,7847,1)  
val.describe()
```

Show hidden output

```
from sklearn import preprocessing  
scaler = preprocessing.StandardScaler().fit(val)  
dvs = scaler.transform(val)  
dvs=pd.DataFrame(dvs)  
dvs.columns=[0,2,4,5,7,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,30,38,39,40,42,43,45  
dvs.describe()
```

Show hidden output

```
VMv = pd.DataFrame(dvs,columns=[0,23,24,25,30,38,39,45,46,47,52,55,56,57,58,59,60,61,62,63])  
Xv = VMv  
Rv = dvs[[30,43,51]]
```

```
VMv.index
```

Show hidden output

```
### Calculando as predições a partir dos modelos elaborados
```

```
ypred = lr.predict(Xv)  
yr_pred = res.predict(Rv)
```

```
ypred = ypred + yr_pred #correção com modelo de resíduos
```

```
labb.index = range(0,7847,1)
```

```
i=0  
bias=0
```

```
while i in range(7847):  
    if labb[i]!=0:  
        bias=(labb[i]-ypred[i])*0.7 + bias*0.3  
    else:
```

```
bias=bias
ypred[i]=ypred[i]+bias
i=i+1
```

```
labb.describe()
```

Show hidden output

```
results = pd.DataFrame(ypred)
results[1] = ytest
#remoção de outliers
results.drop(results.loc[results[0]<40].index, inplace=True)
results.drop(results.loc[results[0]>64].index, inplace=True)
ypred = results[0].values
ytest = results[1].values
```

```
plt.figure(figsize=(25, 10), dpi=80, facecolor='w', edgecolor='k')
plt.style.use('seaborn-whitegrid')
plt.plot(results[0], '-s', label = 'Predição dados de validação',linewidth=1.5,c = 'magenta')
plt.plot(labb, 'o', label = 'Dados', c='black')
plt.tick_params(axis='x', which='major', labelsize=15)
plt.tick_params(axis='y', which='major', labelsize=15)
leg = plt.legend(loc='upper right', fontsize = 17)
leg.get_frame().set_alpha(0.5)
plt.xlabel('Número da Amostra', size = 20)
plt.ylabel('Viscosidade Mooney', size = 20)
plt.axis([0,8000,40,70])
```

Show hidden output

```
results.dropna(how='any',inplace=True)
results
```

Show hidden output

```
ytestf = results[1].values
ypredf = results[0].values
```

```
r2_score(ytestf,ypredf)
```

Show hidden output