UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA QUÍMICA

DENISE MIKI TAWARAYA ISHIDA

PERSPECTIVAS DO APRENDIZADO DE MÁQUINA NO ENSINO DA ENGENHARIA QUÍMICA

São Carlos – SP

2021

DENISE MIKI TAWARAYA ISHIDA

PERSPECTIVAS DO APRENDIZADO DE MÁQUINA NO ENSINO DA ENGENHARIA QUÍMICA

Trabalho de Graduação apresentado ao Departamento de Engenharia Química da Universidade Federal de São Carlos

Orientadora: Profª. Drª. Alice Medeiros de Lima

São Carlos – SP

2021

BANCA EXAMINADORA

Trabalho de Graduação apresentado no dia 28 de junho de 2021 perante a seguinte banca examinadora:

Orientadora: Profª. Drª. Alice Medeiros de Lima, DEQ/UFSCar

Convidada: Drª. Simone de Carvalho Miyoshi, DEQ/UFSCar

Professor da Disciplina: Prof. Dr. Paulo Waldir Tardioli, DEQ/UFSCar

AGRADECIMENTOS

Primeiramente, aos meus pais que trabalharam muito para me oferecer diversas oportunidades e são uma inspiração para mim.

Aos meus irmãos, Juliana Souza e Vinícius Souza que sempre me apoiaram e me motivaram.

À Profª. Drª. Alice Medeiros de Lima, pela oportunidade de ser minha orientadora neste trabalho e por todo o aprendizado.

A todos os professores da UFSCar que contribuíram para a minha formação acadêmica.

Ao Flávio Leal, que sempre esteve ao meu lado compartilhando as conquistas e me oferecendo suporte nos momentos difíceis.

E a todos os meus amigos e familiares que me acompanharam e me deram força para seguir em frente.

RESUMO

O aprendizado de máquina é uma área da Inteligência Artificial que consegue prever resultados a partir de um grande conjunto de dados. Desta forma, muitas áreas já se beneficiam dessa nova tecnologia como, por exemplo, empresas de análise de crédito que utilizam essa ferramenta para detecção de fraudes, serviços de entretenimento para um sistema de recomendação e na medicina para diagnóstico de doenças. Na Engenharia Química, o aprendizado de máquina ainda não é amplamente utilizado, porém existem alguns estudos com aplicações na área que demonstram ser uma ferramenta com grande potencial para a área. Este trabalho constitui uma revisão bibliográfica para entender o que é machine learning, como funciona esse aprendizado, as principais ferramentas e exemplificar possíveis aplicações do aprendizado de máquina na Engenharia Química e frisar sua importância. Dado este potencial, é interessante abordar o assunto no ensino de Engenharia Química para que os alunos possam ter maior contato com o aprendizado de máquina e aplicar esse conhecimento na indústria. Com esse intuito, este trabalho sugere algumas mudanças na grade curricular dos estudantes de Engenharia Química da Universidade Federal de São Carlos para incluir o aprendizado de máquina no percurso formativo.

Palavras-chave: Aprendizado de Máquina. Ensino de Engenharia Química. Inteligência Artificial.

ABSTRACT

Machine learning is an area of Artificial Intelligence that can predict results from a large set of data. Therefore, many areas are already benefiting from this new technology, such as credit analysis companies that use this mechanism for fraud detection, entertainment services for a recommendation system and in medicine for disease diagnosis. Currently, machine learning is not very often used in Chemical Engineering, but there are some studies with applications in the area that demonstrate that it can be a tool with great potential for Chemical Engineering. This work is a bibliographical review to understand more about machine learning, what is the learning process, the main tools and studies to exemplify these possible applications of machine learning in Chemical Engineering and emphasize its importance. Because of this potential, it is interesting to approach the subject in the learning of Chemical Engineering so students can have greater contact with machine learning and apply this knowledge in the industry. With this in mind, this work suggests some changes in the curriculum of Chemical Engineering students at Universidade Federal de São Carlos to include machine learning in the Chemical Engineering education.

Keywords: Machine Learning. Chemical Engineering Education. Artificial Intelligence.

LISTA DE FIGURAS

LISTA DE TABELAS E QUADROS

SUMÁRIO

1 INTRODUÇÃO

Por volta de 1999, houve uma mudança no cenário tecnológico em que se iniciou a integração de sistemas e objeto com a própria internet chamado de Internet das Coisas (COELHO, 2016), sendo possível gerar e coletar suas informações. Essa integração permitiu que os dados desse objeto pudessem ser gerados em grande quantidade, armazenados e coletados. Somente a coleta de dados pode não ter nenhum significado, mas pode gerar informação e conhecimento para trazer melhorias nas mais diversas áreas ao processar e analisar esses dados. Uma das formas de processar e analisar esses dados é com o aprendizado de máquina.

Em 1950, houve o surgimento das Inteligências Artificiais em que as máquinas simulam a inteligência humana (GOMES, 2010). O aprendizado de máquina é uma ramificação da Inteligência Artificial que utiliza uma grande quantidade de dados para simular o aprendizado humano. Assim como o os humanos aprendem com experiência de vida, a máquina aprende a partir dos dados, conseguindo observar padrões e prever resultados. A grande vantagem do aprendizado de máquina é que não é necessário conhecer o caminho entre o dado de entrada e o dado de saída em que seria realizado um estudo por trás para descobrir a matemática que envolve esses dados e que provavelmente necessitaria de um grande esforço e tempo.

Assim, a possibilidade de extrair novos conhecimentos a partir de uma grande quantidade de dados fez com que muitas áreas se interessassem no assunto e nas suas aplicações para a melhoria dos negócios como detecção de fraudes para análise de crédito, sistemas de recomendação para usuários em aplicativos de entretenimento e diagnósticos médicos na área da saúde (MOHRI, 2012). Nessas áreas, o aprendizado de máquina é mais desenvolvido e as empresas já utilizam em seus negócios. Porém, outras áreas ainda não são tão desenvolvidas e não aproveitam a ferramenta amplamente, como é o caso da indústria química.

Na indústria, o aprendizado de máquina pode auxiliar na otimização de processos e redução de custos com embasamento nos dados gerados. Apesar de ainda não ser amplamente utilizado, existem estudos com possíveis aplicações principalmente nas áreas de desenvolvimento de materiais, estimativa de processo e previsão de falhas

(CARTWRIGHT, 2020). Além dos estudos, algumas indústrias químicas começaram a aplicar o aprendizado de máquina para a melhoria de seus negócios no processo de produção que reforçam o indício de que será um assunto de relevância para o futuro.

Apesar de apresentar potencial na área, o aprendizado de máquina no ensino de Engenharia Química ainda não é muito explorado, porém é interessante que os alunos tenham maior contato e conhecimento do assunto para que os alunos possam identificar e avaliar oportunidades de aplicação em suas vidas profissionais.

Este trabalho é um estudo sobre o aprendizado de máquina, um assunto que vem crescendo nos últimos anos e vem mostrando potencial na área da Engenharia Química. O objetivo do trabalho é compreender o aprendizado de máquina, verificar o potencial na Engenharia Química e sugerir como é possível incluir o aprendizado de máquina no ensino. Para isso, será realizado uma detalhada revisão da bibliografia que contemplará o aprendizado de máquina, introduzindo o contexto do seu surgimento, definição, principais tipos, como funciona um aprendizado de máquina e por fim, quais são as principais ferramentas utilizadas para auxiliar. Após a compreensão do que é e seu funcionamento, será apresentado alguns estudos que aplicam o aprendizado de máquina em problemas da Engenharia Química, bem como algumas indústrias que começaram a implementar o aprendizado de máquina em sua produção para demonstrar suas possibilidades na área. Assim, será apresentado com os seguintes tópicos:

- O que é aprendizado de máquina?
- Ferramentas de aprendizado de máquina
- Aplicações na Engenharia Química

Por fim, será proposto como o aprendizado de máquina pode ser inserido na grade curricular do estudante de Engenharia Química da Universidade Federal de São Carlos. A proposta será baseada nos principais assuntos apresentados durante a revisão bibliográfica e sobre pontos importantes para que o aluno compreenda o aprendizado de máquina e seja capaz de aplicá-lo.

2 REVISÃO BIBLIOGRÁFICA

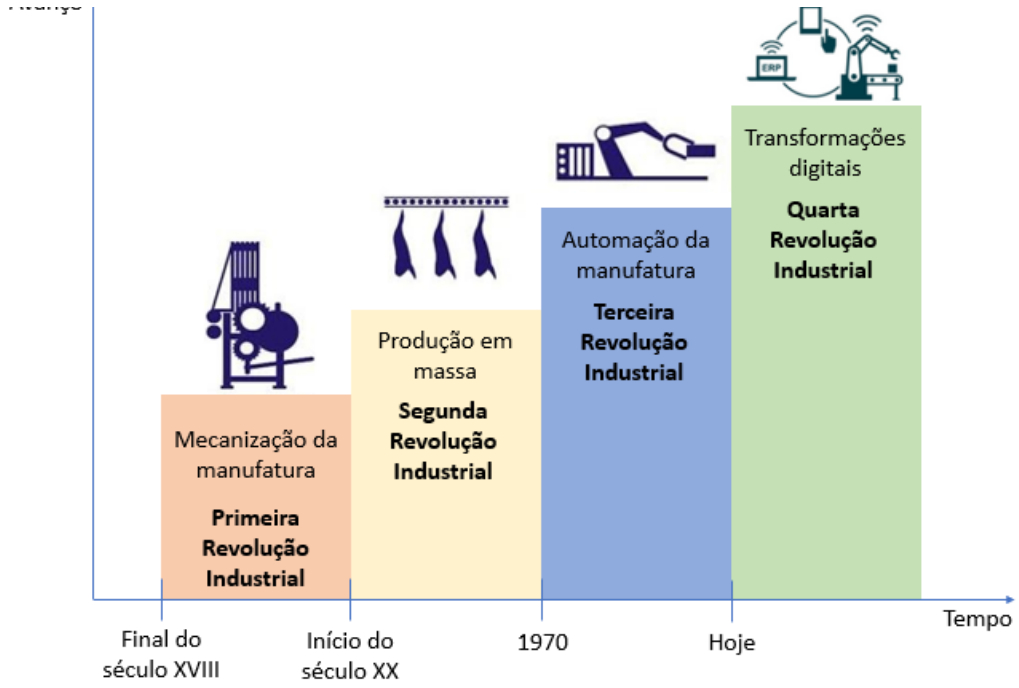
Neste capítulo, será discutido sobre o que é aprendizado de máquina, em qual contexto foi inserido, quais os principais tipos de aprendizado, como é feito o aprendizado e ferramentas para o aprendizado.

Após compreender os principais pontos sobre aprendizado de máquina, será abordado como pode ser aplicado na Engenharia Química e sugestões para grade curricular de estudantes de Engenharia Química voltadas para o aprendizado de máquina.

2.1 O que é aprendizado de máquina?

Conforme a tecnologia vai avançando, os meios de produção vão sofrendo transformações para acompanhar essa evolução. A Primeira Revolução Industrial ocorreu com o surgimento do tear mecânico, que foi sendo aperfeiçoado com máquinas a vapor e possibilitou que a produção passasse a ser feita por máquinas e não mais de forma manual como era feito anteriormente. No século XIX, o aumento da produção de aço resultou em máquinas mais modernas, que junto com o uso da energia elétrica, houve o impulsionamento da manufatura em massa, o que caracterizou a Segunda Revolução Industrial. Já no século XX, tivemos o surgimento do Lean Manufacturing, sistema de produção desenvolvido pela Toyota, em que se tem como princípio o desperdício mínimo e na qualidade do produto e do processo produtivo e o avanço na área da automação e tecnologia da informação (TI) foram grandes responsáveis por esses aprimoramentos (SACOMANO, 2018). Esse período marcou a Terceira Revolução Industrial. E agora no século XXI com o surgimento da internet, temos mais uma transformação ocorrendo na indústria. A Figura 1 ilustra esses avanços tecnológicos que transformaram os meios de produção no passar do tempo.

Figura 1 - Evolução da revolução industrial



Fonte: Elaboração própria (adaptado de Nunez, 2015)

Ainda no século XX, existia o desejo de integrar remotamente as operações industriais com fornecedores e clientes, mas o custo era alto e faltava capacidade dos equipamentos. Desta forma, a indústria impulsionou o desenvolvimento de softwares e hardwares que foram capazes de ultrapassar essas dificuldades e iniciar o processo de integração. Assim em 2011, o governo da Alemanha lançou um projeto durante a Feira de Hannover, denominado Plataforma Indústria 4.0, para fazer com que os sistemas automatizados que controlam equipamentos industriais pudessem trocar dados e informações entre humanos e máquinas para otimizar o processo de produção (SACOMANO, 2018), introduzindo o termo “Indústria 4.0” para essa Quarta Revolução Industrial.

Coelho (2016) destaca que a Indústria 4.0 tem como principais pilares a internet das coisas (*Internet of Things* - IoT) e serviços (*Internet of Services* - IoS); sistemas cyber-physical(CPS) e Big-Data. Pereira (2018) define Cyber-Physical Systems (CPS) como os componentes que integram o mundo físico ao virtual, equipamentos que armazenam dados

sobre o seu estado e realizam operações. A internet das coisas refere-se à conexão de objetos que tradicionalmente operam de forma independente à internet para que possam operar sinergicamente. Já a internet dos serviços vai além da internet das coisas, pois não só conecta objetos à internet, mas também é atribuído um serviço a ele, por exemplo um galão de água conectado à internet que mostra a medição do nível de água no celular e quando o nível chega a 10% da capacidade, envia um pedido de outro galão para a distribuidora (MENA, 2018). Com o surgimento da internet das coisas e serviços, houve um aumento dos objetos que estão ligados a internet e consequentemente, também houve o aumento de dados gerados. Big Data se refere ao imenso volume de conjuntos de dados que alcançam elevadas ordens de magnitude(volume), com variedade e gerados em tempo real, que já não era possível de ser processados com as tecnologias tradicionais de bancos de dados (INTEL, 2013).

Dados, informação e conhecimento são importantes ferramentas para que as decisões sejam tomadas com rapidez e qualidade. Os dados são uma forma de matéria-prima, que ao serem processadas, geram informações capazes de auxiliar na tomada de decisões. O conhecimento é a informação processada pelos indivíduos, sendo o grande desafio transformar os dados em conhecimento (ANGELONI, 2003). Assim, o real valor do Big Data não é só no volume, variedade e velocidade que é produzido, mas sim o conhecimento que ele produz quando analisado — buscando padrões, derivando significado, tomando decisões e, por fim, respondendo ao mundo com inteligência (INTEL, 2013). Uma das formas que a tecnologia proporciona para processar os dados em conhecimento é com o aprendizado de máquina, também conhecido como *machine learning*.

Damos o nome de máquina para o computador, que é um processador de dados, ou seja, um transformador de dados iniciais (dados de entrada) em dados finais (dados de saída). Assim, o aprendizado de máquina é o aprendizado feito pelo computador. A Figura 2 ilustra essa transformação dos dados em um computador.

Figura 2 - Processamento de dados

Fonte: Elaboração própria (adaptado de Gotardo, 2015)

O aprendizado de máquina é similar com o que acontece com os humanos. Para nós, a experiência da vida é o que nos traz o aprendizado. Para as máquinas, os dados fornecidos são como a experiência de vida, são as informações necessárias para a máquina experimentar o problema inúmeras vezes até aprender.

De acordo com Mohri (2012), aprendizado de máquina pode ser definido como métodos computacionais que usam a experiência para o aprendizado, para melhorar o desempenho ou fazer previsões através de algoritmos. A máquina não precisa de uma fórmula matemática para retornar quais serão os dados de saída, pois a partir dos dados fornecidos ela é capaz de gerar as próprias regras a serem usadas para deduzir o resultado de um determinado ponto amostral. Assim, não é necessário ter o conhecimento do caminho entre a entrada e a saída, sendo os dados a experiência necessária para que a máquina aprenda com eles qual é a resposta. Algumas aplicações do aprendizado de máquina são detecção de fraude, mecanismo de recomendação, análise de comportamento de usuários entre muitas outras.

Existem termos que são comumente utilizados no campo de aprendizado de máquina. De acordo com Mohri (2012), as definições mais comuns são:

- Exemplo: Uma linha do conjunto de dados coletados utilizados para aprendizagem ou avaliação, ou seja, os dados de entrada e saída de um caso que servirá para o aprendizado.
- *Features*: Um atributo ou característica dos exemplos que que apresentam correlação com a variável alvo, e, portanto, podem ajudar nas previsões.

- Rótulos: Valores ou categorias em que os conjuntos de dados são divididos, podendo também ser definidos como os dados de saída que se está tentando prever. Em um problema de classificação, são as categorias que o exemplo se encaixa e já para o problema de regressão, é o valor numérico.
- Hiperparâmetro: Parâmetros livres que não são determinados pelo algoritmo, mas sim um dado de entrada definido previamente pelo usuário.
- Amostra de treinamento: Exemplos utilizados para a fase de treinamento do algoritmo.
- Amostra de validação: Exemplos utilizados para a fase de validação do algoritmo. A fase de validação é utilizada para selecionar valores apropriados para os hiperparâmetros.
- Amostra de teste: Exemplos utilizados para testar o algoritmo com outros dados e validar se consegue fazer previsões com dados além do de treinamento.
- Erro: É a função que mede a diferença entre o valor previsto e o valor real, sendo assim uma forma de verificar se o modelo consegue fazer boas predições ou não.
- Conjunto de hipóteses: Uma hipótese é um dos candidatos ao modelo que mais se adequa ao problema, que pode mapear os dados de entrada com os dados de saída. O conjunto de hipóteses são todas as hipóteses possíveis que a escolha do algoritmo pode prover. Será verificado qual a melhor hipótese que se encaixa com o conjunto de dados com a função erro.

Outra definição muito importante é o algoritmo. Para o aprendizado de máquina, o algoritmo é um código de programação computacional com um roteiro lógico que é executado para o conjunto de dados até que um determinado critério de verificação seja atendido, ponto em que a máquina gerou um modelo capaz de prever os resultados, dado qualquer ponto amostral. Uma definição de algoritmo, de acordo com Medina (2005), é um procedimento do passo a passo de ações a serem executadas para realizar alguma tarefa. A vantagem do aprendizado de máquina é justamente a grande capacidade e velocidade do computador com o algoritmo, que pode testar milhões de vezes para prever o modelo.

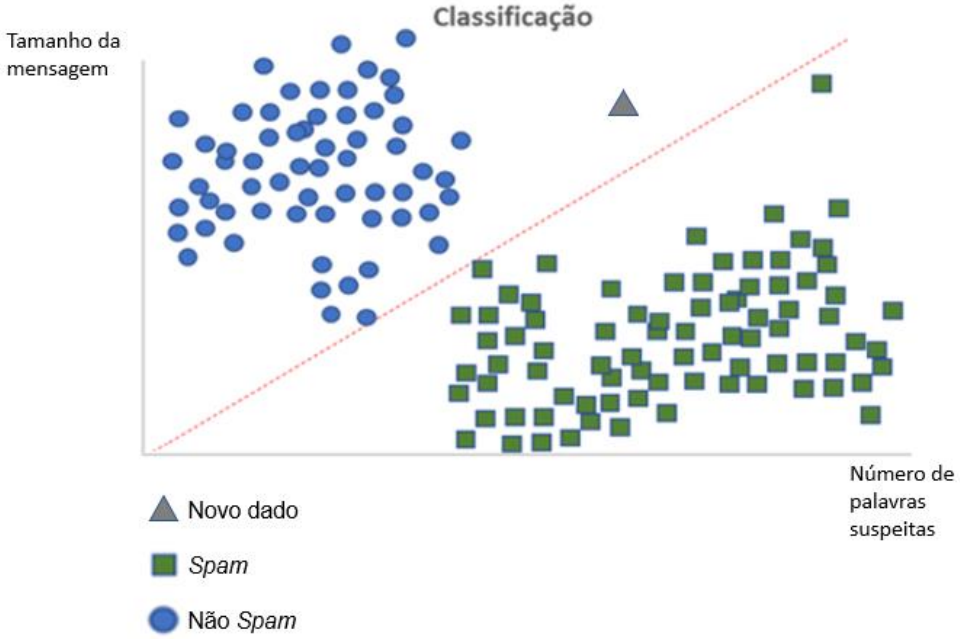
2.1.1 Tipos de aprendizado de máquina

Existem vários procedimentos para que as máquinas aprendam com os dados e são classificados em tipos de aprendizado. Os principais tipos de algoritmos para aprendizado de máquina são o aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (BONACCORSO, 2017).

No aprendizado supervisionado, um conjunto de dados de entrada e saída é utilizado para gerar um modelo capaz de prever os dados de saída para qualquer dado de entrada de forma a minimizar o erro. Após o treinamento, a máquina consegue fornecer os dados de saída com outros dados de entrada que nunca foram vistos (MARSLAND, 2015). Esse tipo de aprendizado é utilizado quando os dados preveem algum evento ou número.

Os algoritmos de aprendizado supervisionado podem ser divididos em duas categorias de acordo com o tipo de dado que será previsto. Pode ser aprendizado supervisionado de classificação, se os valores previstos são um conjunto de categorias pré-definidas(rótulo) como, por exemplo, para fazer a predição se um e-mail recebido é um spam ou não, alocando o dado de entrada(e-mail) em um desses dois rótulos (spam e não spam). Ilustrando esse exemplo na Figura 3, os dados de entrada já identificados se são spam(quadrado verde) ou não spam(círculo azul) são alocados conforme o tamanho da mensagem e o número de palavras suspeitas. O aprendizado supervisionado de classificação cria um modelo para rotular novos dados, representado pela linha tracejada vermelha que divide os casos *spam* e não *spam*. Então após o aprendizado, um novo dado inserido (triângulo cinza) acima da linha tracejada vermelha conforme o tamanho da mensagem e número de palavras suspeitas e é previsto como não-*spam.*

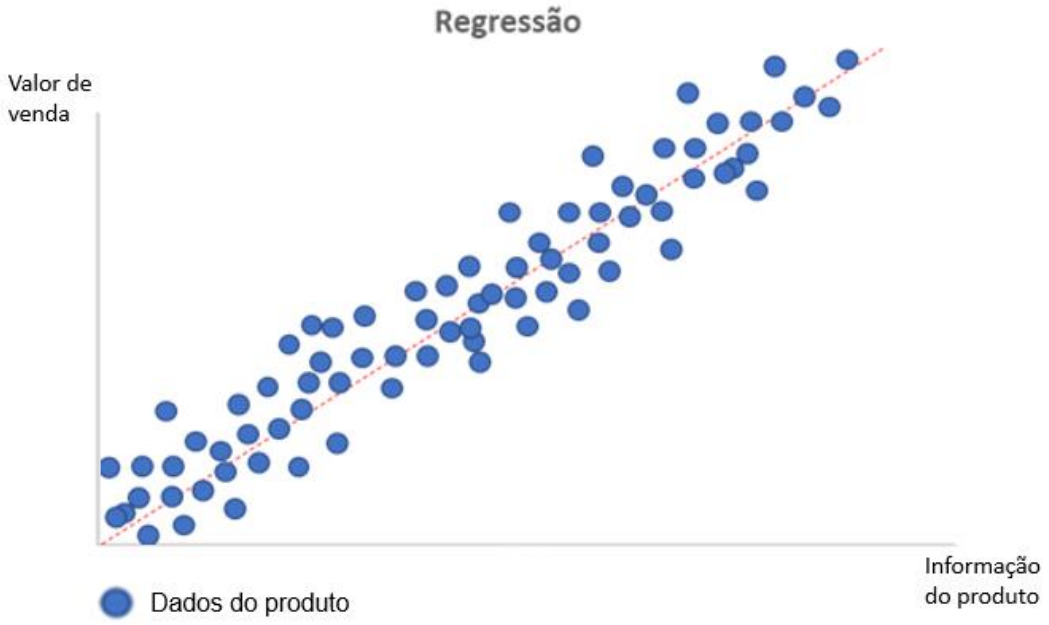
Figura 3 - Aprendizado supervisionado de classificação



Fonte: Elaboração própria

O aprendizado supervisionado também pode ser de regressão, caso os rótulos sejam valores numéricos (NASTEK, 2017). Na Figura 4 é ilustrado um exemplo de regressão para realizar a previsão do número de vendas de um produto. São fornecidas as informações do produto (dados de entrada) com os valores de vendas do produto correspondente (dados de saída), apresentados nos círculos azuis. Assim, o aprendizado supervisionado de regressão cria um modelo, representado pela linha tracejada vermelha, que tenta prever a partir das informações do produto quais serão os dados de saída. Então após o aprendizado, com o dado de entrada é possível usar o modelo para prever qual será o valor de venda.

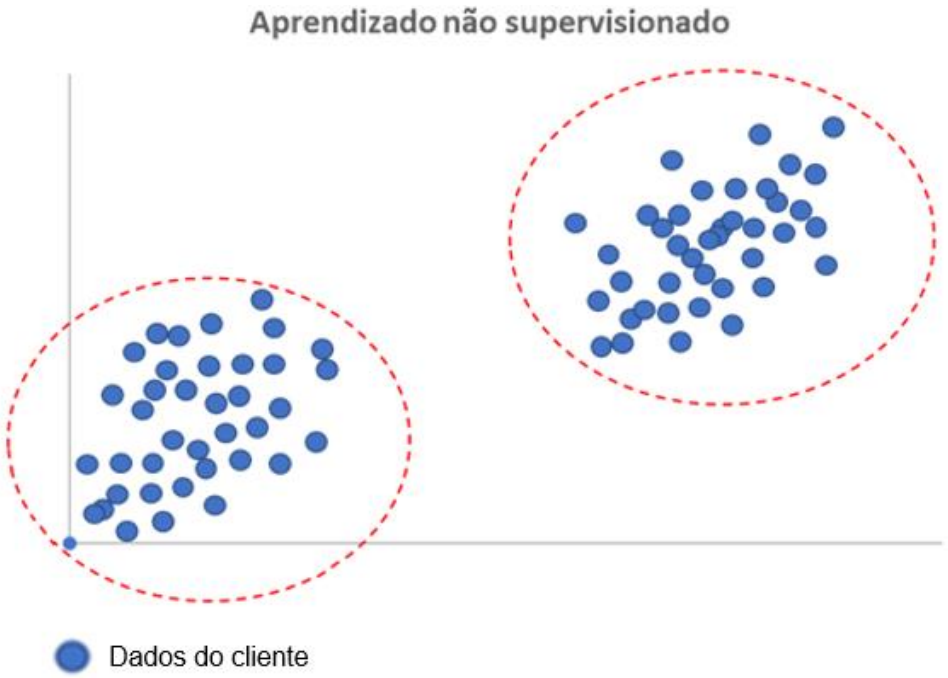
Figura 4 - Aprendizado supervisionado de regressão



Fonte: Elaboração própria

Outro tipo de aprendizado é o aprendizado não supervisionado, em que não é fornecido os dados de saída esperados para validar e comparar o resultado previsto pelo modelo como é feito no aprendizado supervisionado. Neste caso, a máquina identifica as semelhanças e tendências entre os dados e os categorizam, dividindo em grupos que possuem essas semelhanças (MARSLAND, 2015). É uma estratégia muito utilizada em marketing, por exemplo, para identificar clientes com comportamentos semelhantes e segmentar o público-alvo para enviar uma propaganda mais direcionada a esses clientes e aumentar as chances dessas pessoas comprarem o produto. Na Figura 5, temos as características dos clientes, representadas pelos círculos azuis, sem serem identificados previamente e o aprendizado não supervisionado mapeia e identifica diferentes grupos que possuem características semelhantes, limitados pela linha tracejada vermelha. Alguns tipos de aprendizado não supervisionado são clusterização, k-*mean*, análise hierárquica de cluster.

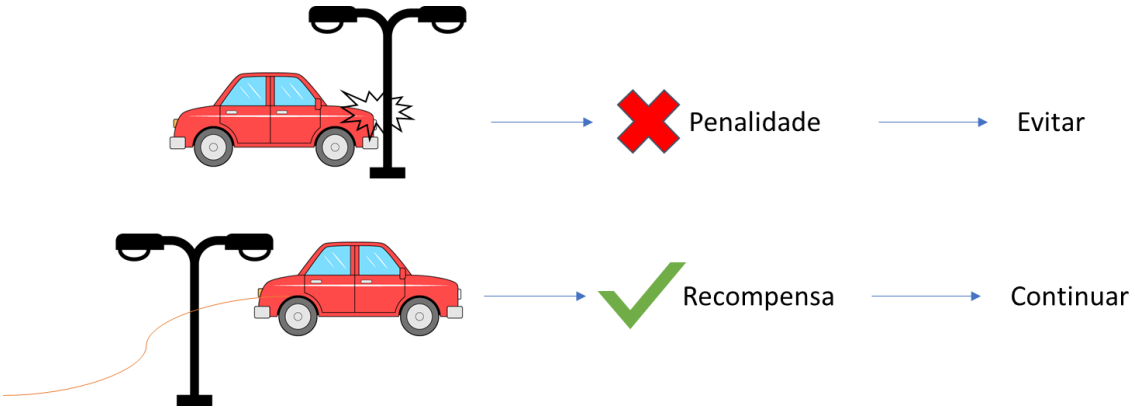
Figura 5 - Aprendizado não supervisionado



Fonte: Elaboração própria

Por fim, no aprendizado por reforço, o algoritmo interage com o ambiente e é informado se está certo ou errado com recompensa ou penalidade e aprende repetindo os acertos e evitando os erros. Muito similar quando treinamos um cachorro para fazer algum truque, em que é dado uma recompensa quando se faz o truque certo. Assim, a máquina experimenta diversas vezes até chegar na resposta certa (MARSLAND, 2015). Um famoso exemplo do aprendizado por reforço é o programa *AlphaGo,* que foi aplicado o algoritmo para aprender a ganhar o jogo *Go,* em que inicialmente foi analisado milhões de jogos e depois o programa jogou contra si mesmo até aprender o caminho da vitória e ganhou do campeão mundo de *Go*, Lee Sedol (HASSABIS, 2016). Para ilustrar, temos a Figura 6 com um exemplo de carro que dirige sozinho. Para esse aprendizado por reforço, caso o carro colida recebe uma penalidade que faz com que a máquina evite repetir a ação. Se o carro conseguir percorrer o percurso, é recompensado e entende que deve continuar a reproduzir a ação tomada.

Figura 6 - Aprendizado por reforço



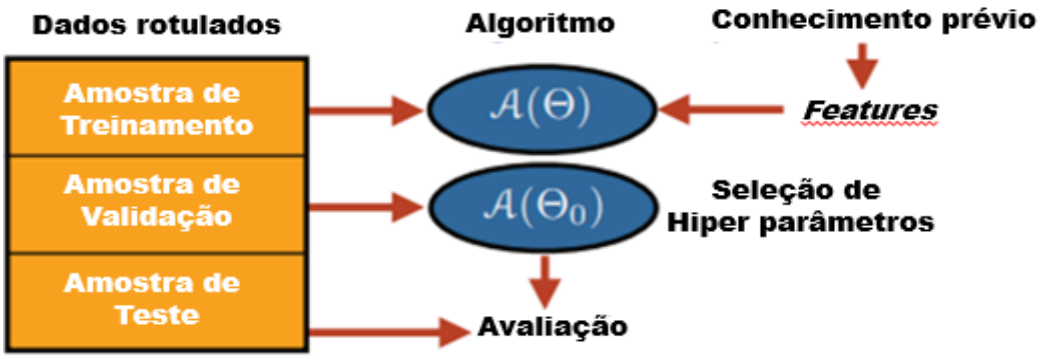
Fonte: Elaboração própria

Não existe um algoritmo melhor ou pior que o outro, mas sim um que se adequa melhor a cada propósito. Então, a escolha do algoritmo é um passo muito importante para o aprendizado e deve ser feito tendo em mente o objetivo e os tipos de dados que a situação apresenta.

2.1.2 Processo do aprendizado

Existem diversos passos para o aprendizado de máquina como coletar os dados, prepará-los e verificar no final do processo se a máquina é capaz de fazer as predições dos dados. A Figura 7 representa esses passos que serão descritos a seguir. De acordo com Marsland (2015), o primeiro passo é a coleta de dados. É importante que tenha uma grande quantidade de dados e que eles sejam de qualidade, ou seja, precisam ser relevantes com as informações necessárias. Não é uma tarefa simples pois pode requerer muitas medições, ilustrando como o avanço do *Big Data* é um marco importante para o aprendizado de máquina. Vale ressaltar que no caso do aprendizado supervisionado, além dos dados de entrada também é necessário exemplos que contenham valores esperados. No caso da detecção se um e-mail recebido é um *spam* ou não, os dados coletados seriam uma coleção de e-mails que servem como exemplos para a máquina.

Figura 7 - Etapas do processo de aprendizado de máquina



Fonte: Elaboração própria (adaptado de Mohri, 2012)

Depois de coletado, os dados devem ser aleatorizados para que a ordem não influencie no aprendizado. Por exemplo, se em um aprendizado de classificação os seus dados estiverem ordenados em todos da categoria A e depois da categoria B, essa ordem pode tornar um fator para a máquina identificar em qual categoria pertence. Os dados aleatorizados serão divididos em amostras de treinamento, validação e teste (MOHRI, 2012).

Para guiar o algoritmo e facilitar o aprendizado, o usuário deve selecionar *features* que estão relacionadas com o problema. O usuário deve usar o conhecimento que tem do problema para escolher variáveis que apresentam correlação com o rótulo a ser previsto. A escolha dessas *features* deve ser cautelosa porque assim como elas podem ajudar o algoritmo, escolher variáveis erradas também podem confundi-lo (MARSLAND, 2015). Na detecção de spam temos como possíveis características o remetente, tamanho da mensagem, palavras-chaves na mensagem.

Dado a existência de diversos algoritmos que foram criados e o conjunto de dados, é necessário fazer a escolha do algoritmo mais apropriado para o problema. Diversos fatores podem ser considerados como o custo computacional de treinamento, a necessidade de interpretar, performance, custo de desenvolvimento, existência de dados rotulados, possibilidade de aplicação de métodos de reforço e formato dos dados de entrada e saída. Como dito anteriormente, o problema de detecção de *spam* envolve tentar prever se

o e-mail é um spam ou não e o retorno é um dos rótulos definido previamente (*spam* ou não-*spam*). Então neste caso, um algoritmo de aprendizado supervisionado de classificação seria o algoritmo mais adequado. Além disso, para alguns algoritmos deve ser configurado manualmente alguns hiperparâmetros de entrada ou requerem uma experimentação prévia para identificar os melhores valores, onde entram as amostras de validação. Esses parâmetros têm a função de que a previsão seja mais precisa (MARSLAND, 2015).

Após selecionado os dados, características, o algoritmo e os hiperparâmetros, começa a etapa de treinamento, sendo trabalho da máquina de utilizar os recursos computacionais e construir um modelo dos dados e fazer a previsão dos dados de saída (MARSLAND, 2015). Nesta etapa, serão utilizados os dados divididos anteriormente e selecionados para a amostra de treinamento. O algoritmo vai selecionando diferentes hipóteses e escolhe o resultado que apresentou melhor desempenho, ou seja, que teve um erro com os valores esperados. O algoritmo vai achar uma função para definir pelas características se o e-mail é *spam* ou não. De início, o algoritmo vai selecionar uma hipótese para tentar dividir esses dados, mas ainda sim terá alguns dados que não irão se encaixar no grupo correspondente. Então o algoritmo vai adaptar essa função (mudando a hipótese) para que esses dados sejam incluídos e assim diminuir o erro.

Infelizmente, nem sempre a máquina será capaz de verificar uma hipótese que se encaixe nos dados e resultará sempre em um erro alto. Neste caso, temos a situação de *underfitting*, em que o algoritmo não é capaz de capturar a dinâmica mostrada pelo conjunto de treinamento (BONACCORSO, 2017). Uma forma de contornar essa situação seria mudar os hiperparâmetros definidos, *features* e se caso continue com o erro alto, recorrer até a troca do algoritmo pois talvez o escolhido inicialmente pode ser muito simples para os dados. Um exemplo, para um modelo de regressão polinomial em que foi feita uma previsão inicial com um polinômio de ordem 1, mas o erro acaba sendo alto pois não se encaixa com os dados fornecidos. Ajusta-se o modelo, mudando a ordem do polinômio e assim, o modelo faz uma previsão com um erro pequeno.

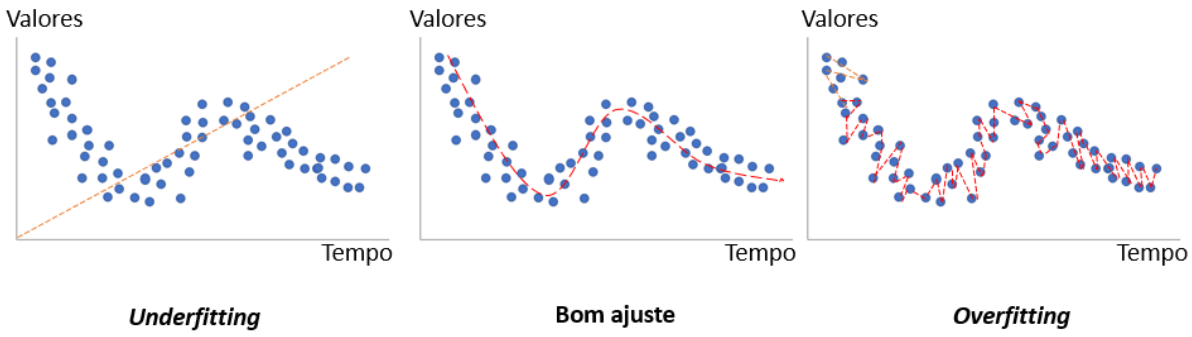
Mesmo a máquina sendo capaz de fazer as previsões, ainda não temos garantias de que as previsões estão corretas. Apesar de minimizar o erro na etapa de treinamento, pode ser que o algoritmo só funcione para o conjunto de dados selecionados para o treinamento. Como exemplo da detecção de spam, o algoritmo foi treinado com os dados de treinamento

coletados anteriormente, mas se chegar novos e-mails pode não ser eficiente. Neste caso ocorre o chamado *overfitting*, quando o algoritmo apresenta uma precisão muito alta com os dados apresentados no treinamento, mas não é capaz de generalizar o problema com dados fora do treinamento (BONACCORSO, 2017).

Para identificar um caso de *overfitting*, pode haver desconfiança caso o erro seja muito baixo, mas pode ser que seja o caso de o modelo ter uma boa previsão. Então, existe um passo de teste em que dados de teste separados anteriormente são utilizados agora. Não será alterada a hipótese definida no passo de treinamento, mas sim será testado e o resultado previsto será comparado com os resultados reais para conseguir identificar como seria a performance da máquina no mundo real pelo erro (MARSLAND, 2015).

Na Figura 8, temos à esquerda um caso de *underfitting* em que o modelo não representa o conjunto de dados e assim apresenta um erro alto. No centro, um caso de bom ajuste em que o modelo se ajusta aos dados apresentados e que consegue ter uma boa previsão. E por fim, à direita, um caso de *overfitting* em que o modelo captura todos os dados apresentados, apresentando um erro muito baixo, mas desta forma não é capaz de ter uma boa previsão com dados além do treinamento.

Figura 8 - Underfitting e overfitting



Fonte: Elaboração própria (adaptado de Bhande, 2018)

Todos os passos ocorrem de forma iterativa, ou seja, caso um passo não ocorra da forma esperada, volta para ajustar algum passo anterior. Por exemplo, na fase de

treinamento, verifica-se que o erro do modelo é muito grande e então volta para o passo de selecionar os hiperparâmetros para fazer ajustes e tentar diminuir o erro.

Assim, depois de todos esses passos, o algoritmo está pronto para fazer previsões com os dados do mundo real. Para manter o desempenho, também é necessário adicionar um passo de monitoramento que verifica se o modelo continua trazendo boas previsões pois pode ter acontecido alguma mudança nos dados de entrada e esse monitoramento pode apontar isso. O aprendizado de máquina é uma ferramenta muito útil, que pode trazer boas predições com base em dados. Por isso, é importante entender o que é o aprendizado de máquina e como funciona para enxergar possibilidades de aplicações em diversas áreas.

2.2 Ferramentas de Machine Learning

Uma ferramenta essencial para o aprendizado de máquina é a linguagem de programação, muito envolvida na etapa de treinamento em que o algoritmo é responsável por construir os modelos utilizando os recursos computacionais. A linguagem de programação é uma linguagem específica que pode dar comandos que o computador consegue compreender e executar e através dela, podendo especificar quais dados um computador vai usar, como estes dados serão tratados, armazenados, transmitidos, quais ações devem ser tomadas em determinadas circunstâncias (GOTARDO, 2015). Então o algoritmo é escrito com as instruções lógicas em uma linguagem de programação(código) e é executado. Existem várias linguagens comumente utilizadas como o Python e o R.

2.2.1 Liguagens de Programação

As linguagens de programação apresentam características distintas entre si e a escolha de qual linguagem utilizar deve ser considerada quais pontos são importantes para a aplicação. Uma característica importante na escolha das linguagens é a legibilidade e facilidade de escrita. A princípio, devido à baixa capacidade computacional das aplicações criadas por equipes pequenas ou individuais e a pequena comunidade global de programadores, o foco estava em linguagens performáticas no lugar da facilidade de programação. Porém, conforme foram decrescendo os custos de processamento e os custos

de manutenção do código ficaram evidentes, cresceu a necessidade de que o código fosse de fácil manutenção, leitura e compreensão. Alguns padrões que foram utilizados para facilitar criação e manutenção de softwares foram:

- a elaboração de sintaxes simples
- automação de tarefas como gerenciamento de alocação de memória, em que não é necessário que seja declarado o espaço que a variável ocupará
- linguagens de tipagem fraca, em que não é necessário declarar o tipo da variável (número inteiro, decimal, texto, etc.)
- aumento na popularidade de linguagens interpretadas, em que o código é executado por um programa (interpretador)
- programação orientada a objetos que consegue separar o programa em partes, facilitando a manutenção e reuso do código

Todos esses pontos facilitam a organização e compartilhamento do código e de serviços permitindo que cada desenvolvedor trabalhe em uma parte diferente do código. Por outro lado, alguns padrões foram removidos ou desincentivados como realizar o mesmo comando dentro de uma mesma linguagem de diversas formas, sobrecarga de operadores sem deixar explícito sua nova função e grandes pedaços de código que realizam diversas funções que podem confundir e dificultar a interpretação da linguagem (SEBESTA, 2011).

Outro fator importante na escolha de uma linguagem é a comunidade que a utiliza. Dentre as linguagens, existem algumas com uma comunidade mais ativa, com um grande número de pessoas que ajudam a trocar informações, tirar dúvidas, compartilhar códigos, encontrar erros nos códigos e verificar com os desenvolvedores para corrigi-los (SARKAR, 2018). Comunidades maiores tendem a gerar projetos *open source*, que não possuem custo de licença, com maior escopo e qualidade. Como a produção de código é compartilhada, a comunidade pode ajudar tanto na testagem como na elaboração de recursos para o projeto. Também é mais fácil encontrar programadores para integrar os times em linguagens mais populares. Linguagens menos utilizadas podem até ser tecnicamente compatíveis com determinado projeto, mas sem uma comunidade forte pode dificultar a execução do projeto.

Além disso, as linguagens de programação apresentam bibliotecas em que se encontram pacotes com funções pré-escritas que podem ser mais facilmente aplicados no código, poupando o trabalho de escrever essa função e deixando o código mais limpo,

resumindo o código em poucas linhas, facilitando a compreensão e diminuindo a possibilidade de erro (MENEZES, 2014). Essas bibliotecas podem ser ferramentas de uma área específica como para o aprendizado supervisionado, estatística, visualização de dados. O que distingue entre as linguagens de programação é a quantidade e variedade de pacotes disponíveis. Uma comunidade forte também pode ajudar com isso para manter as bibliotecas sempre atualizadas, com cada vez mais funções e estabilidade.

Outro ponto importante a se considerar é a flexibilidade da linguagem. Algumas linguagens apresentam um escopo restrito, o que dificulta o desenvolvimento caso o escopo mude ao longo do projeto. Também é importante que seja possível integrar a linguagem com outras plataformas como Excel e MySQL e desse modo, além das ferramentas que a linguagem proporciona, essa integração potencializa a aplicação com as ferramentas das outras plataformas.

Por fim, a performance da linguagem é um fator que voltou a ganhar importância recentemente. Nos últimos anos a velocidade com a qual os hardwares evoluíram diminuiu, enquanto os requisitos para aplicações capazes de lidar com quantidades vastas de dados aumentou. Isso gerou a ressurgência de linguagens altamente performáticas capazes de lidar com esses cenários, sem sacrificar a velocidade e facilidade de desenvolvimento. Apesar de ainda não afetar a comunidade de ciência de dados, é possível que essas limitações moldem o futuro das linguagens utilizadas para aprendizado de máquina.

2.2.2 Escolha da linguagem de programação

Foram citadas acima algumas características que devem ser levadas em conta para a escolha da linguagem a ser utilizada. Um fator interessante de ressaltar é a importância do entendimento básico da lógica e dos comandos da linguagem do usuário para adaptação e manutenção, por mais que existem algoritmos prontos feitos pela comunidade. Assim, é muito importante fazer uma reflexão sobre qual linguagem aprender ou, caso já exista esse conhecimento prévio, se seria mais vantajoso aprender outra linguagem para a aplicação ou utilizar a linguagem conhecida, pois o processo de aprender uma linguagem nova pode ser complexo e demorado.

Diversas linguagens de programação populares apresentam os requisitos mínimos para serem utilizadas com aprendizado de máquina, entretanto, pelos critérios citados acima, algumas ganharam popularidade e vieram a se tornar referência dentro do campo de ciência dos dados. Python e R são duas linguagens mais notórias enquanto que outras linguagens como Julia e Java também apresentam uso considerável. O nível de conhecimento da linguagem depende do objetivo, se for uma aplicação mais complexa ou não. É necessário ter um conhecimento sobre lógica de programação, mas não é necessário se tornar um especialista em uma linguagem de programação para começar a aplicá-la.

A linguagem mais popular é o Python, que possui uma comunidade grande e ativa de usuários que disponibilizam pacotes eficientes e livres de erro para serem utilizados (SPRINGBOARD, 2020). Isso facilita para quem está começando a aprender e atinge diversos públicos como cientistas, programadores, desenvolvedores. Devido a sua filosofia de manter uma sintaxe simples, mas ao mesmo tempo poderosa e que permita aos desenvolvedores focarem na lógica da aplicação ao invés de implementação técnica, o Python se tornou uma linguagem que rapidamente cresceu em popularidade tanto no campo da ciência de dados, como para desenvolvimento de software (SPRINGBOARD, 2020). Um código escrito em Python por um programador experiente irá se assemelhar a uma série de instruções de comandos lógicos em inglês que torna mais fácil tanto a escrita quanto a compreensão. Esse cuidado na criação de uma sintaxe simples, tornou a linguagem mais acessível para ensinar novos programadores e para facilitar a cooperação entre diversas pessoas.

A popularidade da linguagem resultou em uma ampla gama de projetos de softwares livres focados em criar extensões que facilitem o desenvolvimento na linguagem que resultaram em uma extensa coleção de bibliotecas e pacotes integrados à disposição de qualquer interessado. Essas bibliotecas aumentaram a produtividade dos desenvolvedores ao fornecer formas eficientes de realizar cálculos matemáticos, algoritmos já escritos para serem reutilizados, formas de desenhar gráficos entre outros.

Outro fator que contribui para o uso do Python é que por ser uma linguagem de programação já utilizada para o desenvolvimento de software, ela também se aproveita de diversas funcionalidades desenvolvidas, como integração com diversos tipos diferentes de bancos de dados e APIs. Isso permite uma integração fácil tanto em servidores online ou até

mesmo em pequenos sistemas de placa única como o Raspberry Pi, que é um dos computadores mais simples e de baixo custo. Também facilita a interação entre desenvolvedores de software e cientistas de dados trabalhando em um mesmo produto.

Uma característica frequentemente empregada ao se trabalhar com dados é o fato que Python permite a execução de comandos de forma interativa. Nesse modo, é possível escrever comandos, executá-los, ver o resultado, tudo dentro de uma mesma execução do programa. Outras linguagens que não apresentam essa função, requerem que todo o código seja escrito antes de executar, e caso novos comandos sejam escritos, deve-se executar o código desde o início. Os aplicativos Jupyter Notebooks, Jupyter Labs, Spyder e R studio se aproveitam dessa função para, por exemplo, visualizar o valor de variáveis durante a execução dos programas.

Porém, as funções citadas acima apresentam alguns custos que devem ser considerados. Primeiramente, Python não é uma linguagem que usa recursos de memória e processamento de forma eficiente. Outras linguagens compiladas como Fortran e C apresentam um melhor uso dos processadores e maior facilidade de operar com processadores com multicore. Algumas bibliotecas como Numpy e SciPy podem ser utilizadas para parcialmente contornar esses problemas e melhorar a performance, mas ainda assim não é possível chegar no nível de eficiência de processamento. Além disso, é necessário que se tenha todas as dependências instaladas na máquina para que seja possível rodar efetivamente um código, incluindo a versão do Python e de todas as bibliotecas utilizadas. O uso de versões incorretas pode gerar erros graves e difíceis de serem identificados. Gerenciadores de bibliotecas e ambientes como PIP, que já é incluso na instalação, e Anaconda, focado em ciência de dados, são utilizados para garantir que o código seja executado de forma correta em qualquer máquina. Por fim, o uso intensivo de bibliotecas torna o código dependente de códigos externos que podem não ser atualizados com frequência necessária.

Depois do Python, a linguagem R é considerada como a segunda mais utilizada no campo de aprendizado de máquina e apresenta muitas bibliotecas, principalmente estatísticas e assim, fornece uma variedade de ferramentas para avaliar os algoritmos. Também possui pacotes de aprendizado de máquina integrados e técnicas como visualização de dados, análise de dados e avaliação de modelos. O estilo de programação é fácil e compatível com

outras plataformas e é uma ótima escolha para aprendizado de máquina que usam muitos dados estatísticos (SPRINGBOARD, 2020). A linguagem Java é muito popular no campo da programação em geral e a grande vantagem é que muitas organizações já possuem códigos escritos em Java e pessoas que já utilizavam essa linguagem não precisam aprender uma nova. Possui uma biblioteca de aprendizado de máquina e uma rápida execução. Porém a linguagem não é tão simples quanto Python, é necessário definir tudo de início e isso aumenta o tamanho do código e o tempo de desenvolvê-lo (SPRINGBOARD, 2020). A linguagem Julia é um potencial competição para Python e R devido às ferramentas exclusivas para aprendizado de máquina. Foi especialmente projetada para a implementação de matemática básica e consultas científicas que sustentam a maioria dos algoritmos de aprendizado de máquina. Porém, a linguagem Julia começou em 2009 e por ser relativamente nova, as ferramentas, bibliotecas e pacotes não são tão completos e a comunidade ainda é pequena (SPRINGBOARD, 2020).

2.2.3 Bibliotecas

Por ser uma linguagem com boas ferramentas e muito utilizada para o aprendizado de máquina, Python será exemplificada para citar outras ferramentas mais à frente. Para utilizar uma linguagem de programação interpretada como Python é preciso instalar um programa (interpretador) para que o computador seja capaz de ler o código, interpretar e executá-lo e é possível baixar esse programa no próprio website da linguagem. Caso a linguagem utilizada for uma linguagem executável, não é necessário um interpretador para executá-la. O código consegue conversar diretamente com o computador e executá-lo, porém, é uma linguagem de máquina que é muito mais complexa de ser interpretada por humanos.

A instalação padrão do Python contém uma série de bibliotecas padrões que oferecem um pacote completo para programação orientada a objetos, com estruturas de dados comuns além de várias bibliotecas internas para facilitar o desenvolvimento como suporte para diferentes arquivos e até funções matemáticas. Além disso, é possível instalar bibliotecas necessárias para o projeto que não estão contidas na instalação base usando algum gerenciador de bibliotecas.

Como mencionado anteriormente, Python possui uma coleção de bibliotecas com recursos que podem ser aplicados para áreas específicas como ciência de dados e estatística. Algumas bibliotecas relevantes para o aprendizado de máquina são Numpy, Pandas e Scikit-learn (SARKAR, 2018).

Numpy é uma das mais importantes bibliotecas no Python, muito utilizada para aprendizado de máquina e computação científica pois consegue lidar com cálculos numéricos (SARKAR, 2018). Oferece uma variedade de rotinas para operações em matrizes, incluindo matemática, lógica, álgebra linear básica, operações estatísticas básicas, simulação aleatória e muito mais. Uma grande vantagem da biblioteca é que parte de seu código é compilado em CPython que permite uma execução rápida que utiliza múltiplos processadores.

Para a construção de gráficos e visualização desses dados como gráfico de linha, histograma, distribuição, gráficos 3D e mapa de calor é utilizada a biblioteca Matplotlib. Essa biblioteca foi criada para que utilizasse uma interface semelhante a outra linguagem de programação, o matplot. Através dela é possível criar uma extensa variedade de gráficos, incluindo alguns tridimensionais. Apesar da interface ser de fácil aprendizagem para programadores de matplot, usuários que nunca tiveram contato com a linguagem podem apresentar dificuldades, e devido a isso, foram criadas bibliotecas como o Seaborn que utilizam matplot mas apresentam interface mais amigável. Outras soluções como o plotly vem ganhando popularidade (TRAN, 2020).

Pandas é uma biblioteca importante para preparação e análise de dados (SARKAR, 2018). Seu principal objeto é o DataFrame, que apresenta dados de forma tabular. Apresentam uma ampla quantidade de funções para a manipulação de dados dentro desse formato incluindo filtros, operações entre matrizes, agrupamentos, permite importar dados de diferentes formatos (csv, excel, SQL), fazer alterações nas tabelas como inserir e deletar linhas e colunas específicas, junção de tabelas, plotar gráficos entre outros. Por essa razão, é uma ótima ferramenta para preparação e limpeza de dados, além de permitir manipulação e criação de algoritmos de aprendizado de máquina customizados. Ela usa extensamente bibliotecas como Numpy para melhorar performance e matplotlib para criação de gráficos. Grande parte das bibliotecas de aprendizado de máquina são programadas para consumir dados no formato de DataFrames. Outras ferramentas como pyspark permitem uma interface

semelhante ao do DataFrame para manipulação de dados em bancos de dados Apache Spark para big data, enquanto Dask também apresenta uma interface semelhante, mas permite a análise de dados em situações que a quantidade de dados ultrapassa a memória RAM da máquina utilizada (ZHANG, 2019).

A biblioteca Scikit-learn é uma das bibliotecas mais indispensáveis para o aprendizado de máquina em Python. Nessa biblioteca, encontram-se diversos algoritmos de aprendizado de máquina (SARKAR, 2018) do aprendizado supervisionado e não supervisionado, incluindo as principais áreas como classificação, regressão, clusterização. Também apresenta ferramentas para ajuste de modelo, pré-processamento de dados, seleção e avaliação do modelo e muitos outros. Assim como em Python, outras linguagens também apresentam suas próprias bibliotecas que auxiliam para a construção do código. Porém, vale ressaltar que a linguagem Python possui uma ampla gama de bibliotecas em comparação a outras linguagens e devido à forte comunidade, ela é frequentemente atualizada.

2.2.4 Gerenciadores de ambientes

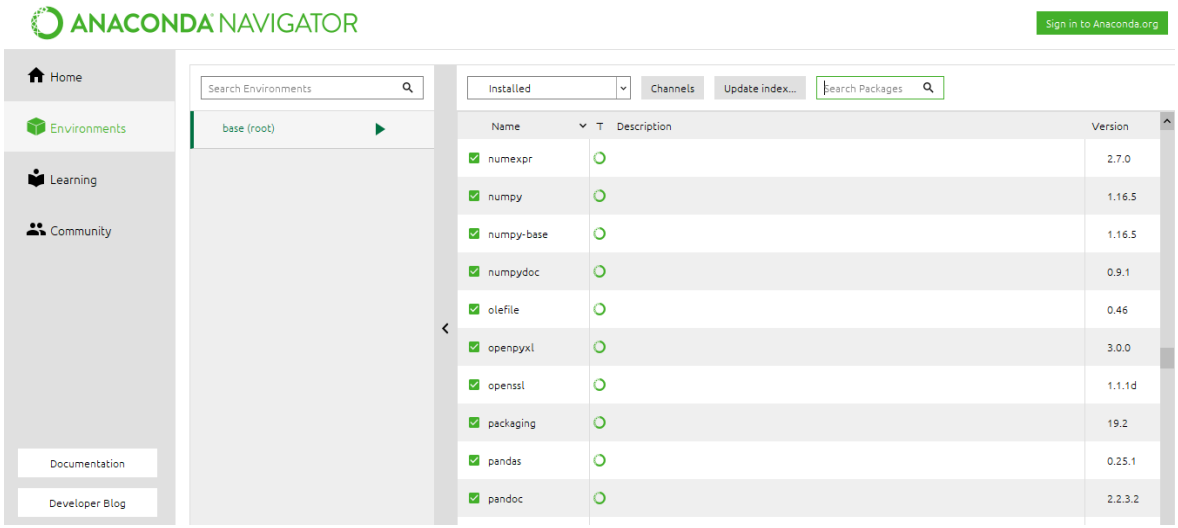
Além das bibliotecas e pacotes temos os gerenciadores de pacotes. São responsáveis por armazenar, disponibilizar e instalar bibliotecas. Através deles é possível a instalação e atualização de bibliotecas para versões específicas. PiP é a ferramenta padrão para instalação de pacotes Python e vem pré instalado junto com as principais distribuições. Com ela, é possível instalar pacotes armazenados no Python Package Index. Outra ferramenta bastante utilizada é o Conda que apresenta tanto uma distribuição *open source* para indivíduos como uma distribuição comercial para empresas. Um diferencial em relação ao PiP possibilidade de além de instalar e gerir bibliotecas Python, também gerir distribuições de Python, R e bibliotecas R.

Existem também os gerenciadores de ambientes, que são responsáveis pelo isolamento de diversas versões de linguagens e bibliotecas dentro de um mesmo sistema operacional. Utilizando esses gerenciadores é possível criar diferentes ambientes, cada uma com suas próprias linguagens e bibliotecas. O Python apresenta sua própria ferramenta padrão de ambientes chamada Venv, entretanto, da mesma forma que o PiP, seu escopo é

apenas para bibliotecas Python. O Conda também pode ser usado como gerenciador de ambientes, e é capaz de gerir versões e bibliotecas das linguagens Python e R.

Por isso, existem compilações de pacotes que instalam no computador tanto o interpretador quanto algumas bibliotecas e não precisar instalar cada biblioteca uma por vez. Um exemplo é Anaconda para Python e R que contém diversas bibliotecas voltadas para ciência de dados e aprendizado de máquina. O pacote de instalação do Anaconda pode ser encontrado no próprio website do Anaconda.

Figura 9 - Bibliotecas instaladas no Anaconda



Fonte: Elaboração própria

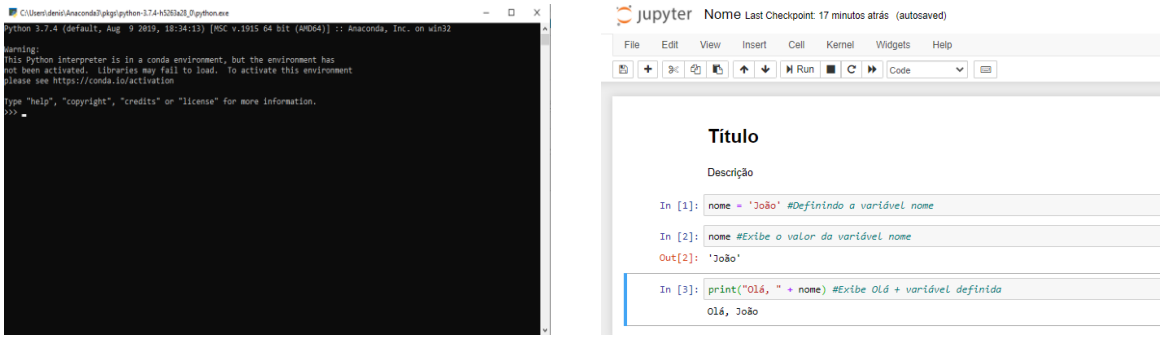
Na Figura 9, temos o Anaconda Navigator, que é uma interface gráfica que permite visualizar e gerir pacotes e ambientes do Anaconda. Pela Figura 9, podemos identificar à direita a lista de gerenciamento de bibliotecas do Anaconda.

Outra ferramenta que auxilia o aprendizado de máquina são os gerenciadores de versão. À medida que os softwares crescem, suas versões são atualizadas e é interessante adotar uma ferramenta para manter a mesma versão para que o projeto não seja perdido por um conflito de versões. Pode se beneficiar dessa ferramenta tanto para evitar essa situação quando há uma atualização ou mesmo para compartilhar um projeto em uma equipe de forma eficiente, pois há a possibilidade de as pessoas terem versões distintas. Utilizando uma

ferramenta de versionamento, diversos usuários podem gerir uma mesma base de códigos, esses sistemas de versionamento apresentam uma ramificação onde é possível realizar diversas alterações em uma cópia do código. Após isso, existe um processo de junção na versão principal do código, onde as suas alterações são incorporadas. A ferramenta mais empregada de versionamento é o GIT, mas existem outras que também podem ser empregadas como Mercurial e Subversion.

Por fim, existem os ambientes, conhecidos como ambiente de desenvolvimento integrado (IDE, do inglês *Integrated Development Environment*), que trazem vantagens em comparação com a janela de terminal, onde seria escrito o código se não fosse utilizado algum IDE.

Figura 10 - Janela de terminal vs IDE



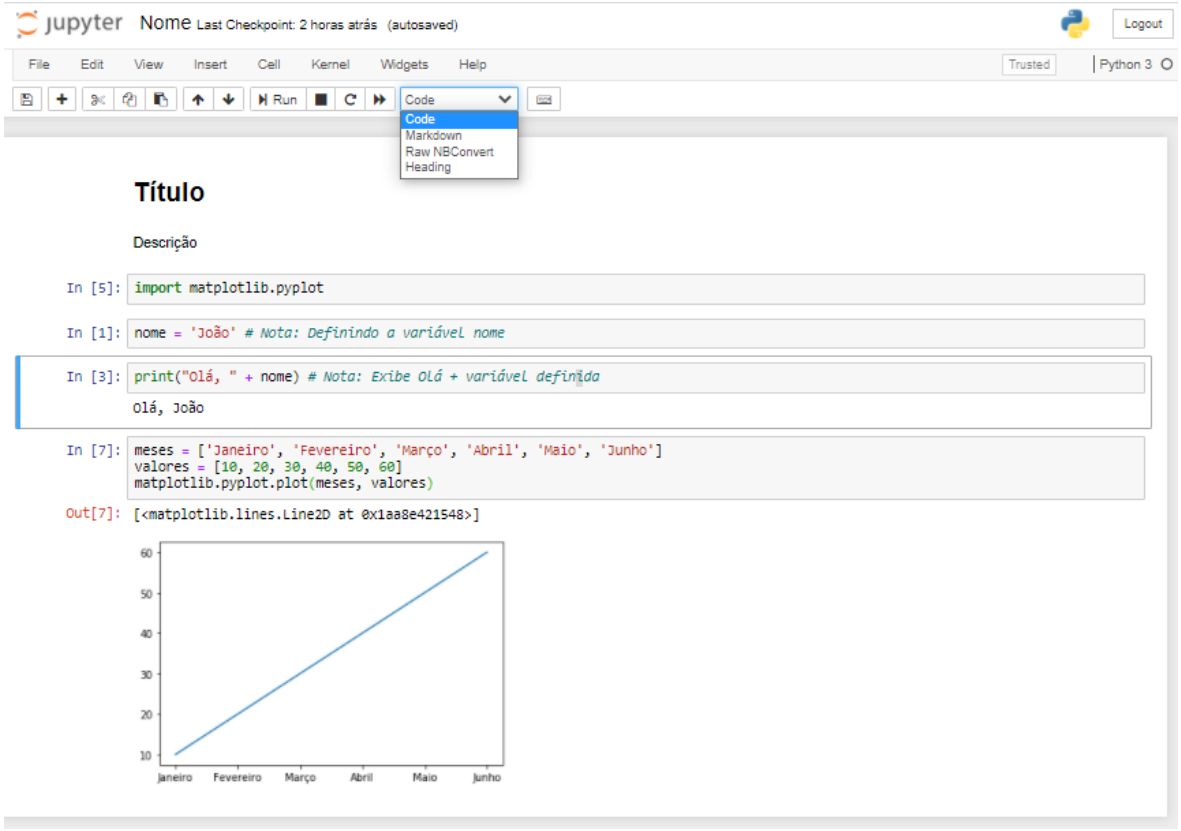
Fonte: Elaboração própria

A Figura 10 demonstra a diferença da janela de terminal e do Jupyter Notebook, uma IDE para Python, em que o Jupyter Notebook possui uma interface mais amigável e organizada. Uma das vantagens que IDE podem trazer é facilitar a escrita do código porque tem uma interface mais visual do que a janela de terminal e, consequentemente, o entendimento do código. Outra vantagem é que podem também trazer resultados intermediários como valores e imagens, que podem ser vantajosos principalmente na área de dados para visualizar melhor os resultados.

O Jupyter Notebook é uma IDE muito utilizado para Python, em que podemos compartilhar, ter um código interativo com notas, imagens, dados de saída e uma interface

visual e amigável, dependendo do que é marcado para o tipo de célula, na parte superior indicado na Figura 11 abaixo. A opção “Code cells”, para escrever códigos e comentários associados e os dados de saída são exibidos como resultado das células; “Markdown” para fazer notas de diversas formas como títulos, textos simples, imagens, equações que podem auxiliar a explicar a lógica do código para quem tiver acesso. Por fim, “Raw NBConvert” exibe o texto do que é escrito, e não são convertidos por mecanismo de conversão dos notebooks, então são sempre mantidos.

Figura 11 - Interface Jupyter Notebook (IDE)



Fonte: Elaboração própria

2.3 Aplicações do aprendizado de máquina na Engenharia Química

O aprendizado de máquina está cada vez mais ganhando importância em diversas áreas como mercado financeiro, tecnológico, varejo e saúde. As previsões feitas com base

em dados auxiliam em tomadas de decisões e planejamento estratégico para essas áreas. Na área da Engenharia Química, é possível melhorar a eficiência e desempenho de um processo químico aplicando o aprendizado de máquina com os dados gerados nesse processo. Algumas empresas como BP, ExxonMobil, HSE, e GE estão investindo em aprendizado de máquina com aplicações para Engenharia Química a fim de melhorar a segurança e confiabilidade do processo, identificar novas conexões e fluxos de trabalho e acelerar os ciclos de vida do projeto (CARTWRIGHT, 2020). Dado o potencial, esse assunto vem cada vez mais sendo explorado na Engenharia Química.

Nos anos 80, os estudos eram mais voltados para os sistemas baseados em conhecimento, regras ou produção conhecido como a "Era dos Sistemas Especialistas", em que os programas de computador imitavam a resolução de problemas provindos de conhecimento prévio de regras heurísticas, ou seja, seguiam as regras de heurísticas com a vantagem de o processamento da máquina ser mais rápido (VENKATASUBRAMANIAN, 2019).

Um exemplo é o estudo feito em 1985, cujo objetivo era realizar a seleção de um método de equilíbrio líquido-vapor adequado, entre três possibilidades e com distintas combinações de equações para cada uma. Essa seleção varia de acordo com as substâncias, pressão, temperatura e concentração e é uma escolha crítica para uma boa simulação de um processo e foi utilizado um sistema especialista para esse estudo. Um sistema especialista é composto por duas partes, o conhecimento prévio e o mecanismo de interferência que determina como o conhecimento prévio será utilizado (BANARES-ALCÂNTARA, 1985). Porém, o sistema especialista apresentou alguns impasses para aplicações na Engenharia Química, principalmente considerando aplicações industriais, já que exigiria muito esforço, tempo e dinheiro e seria difícil e caro manter a base de conhecimento caso ocorresse alguma alteração (VENKATASUBRAMANIAN, 2019).

Após o surgimento do aprendizado de máquina nos anos 90, começaram estudos iniciais sobre as possíveis aplicações na área da Engenharia Química em que não se construía mais o sistema baseado em conhecimento prévio, mas sim o próprio sistema adquiriria o conhecimento com base em uma grande quantidade de dados, o que facilita na manutenção e desenvolvimento de modelos (VENKATASUBRAMANIAN, 2019). Pesquisadores

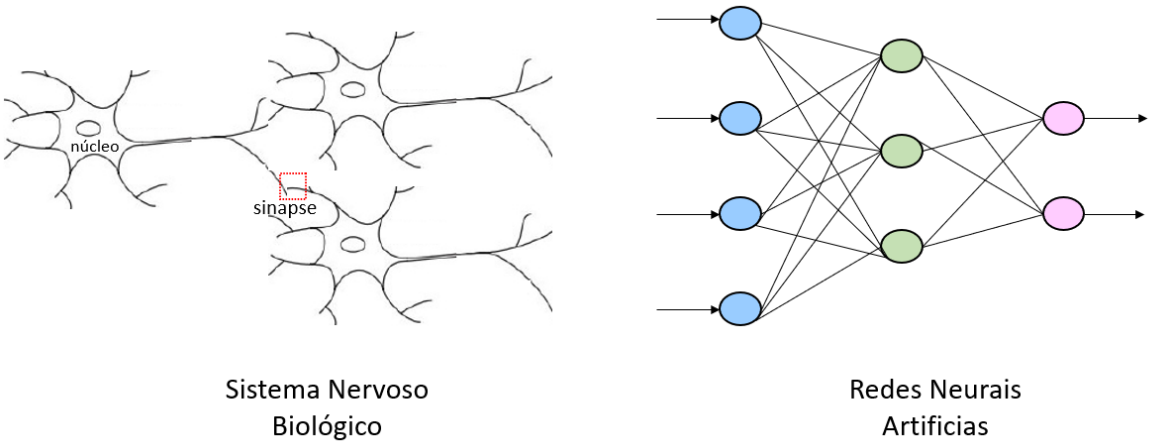
fizeram avanços na Engenharia Química como Azarpour (2017) e Gonzaga (2009) nas áreas de modelagem e controle.

Para a modelagem matemática, a construção de modelo de processo químico é feita pela abordagem mecanicista, baseado nas leis físicas e químicas fundamentais que geralmente envolvem equações matemáticas não lineares com um grande número de variáveis para se trabalhar. Desta forma, é um grande desafio realizar a construção de um modelo de processo químico devido à complexidade de desenvolver os modelos mecanicistas. Além disso, são feitas algumas considerações para algumas propriedades físicas e químicas como idealidade dos gases e linearização de equações não lineares, o que implica algumas limitações do modelo (HAJJAR, 2018).

Considerando isso, o aprendizado de máquina chamou a atenção para essa área devido ao fato que seria possível modelar um processo químico sem a necessidade do conhecimento detalhado e também pode lidar com sistemas mais complexos e não lineares. Um dos métodos utilizados para aplicações em Engenharia Química são as Redes Neurais Artificiais (RNA).

As **Redes Neurais Artificiais** são algoritmos de aprendizado de máquina supervisionado baseado no sistema nervoso biológico, em que o modelo é composto por um grande número de neurônios com várias conexões entre eles que compõem camadas. Como mostrado na Figura 12 à esquerda, o neurônio recebe o estímulo, ocorre o processamento das informações nos núcleos dos neurônios e as informações são transmitidas através das sinapses para outro neurônio. De maneira similar, as Redes Neurais Artificiais recebem as informações nos neurônios de entrada, que processam essas informações e os dados de saída desses neurônios resultam nos dados de entrada para outros neurônios, simulando a sinapse dos neurônios biológicos, ilustrado na Figura 12 à direita.

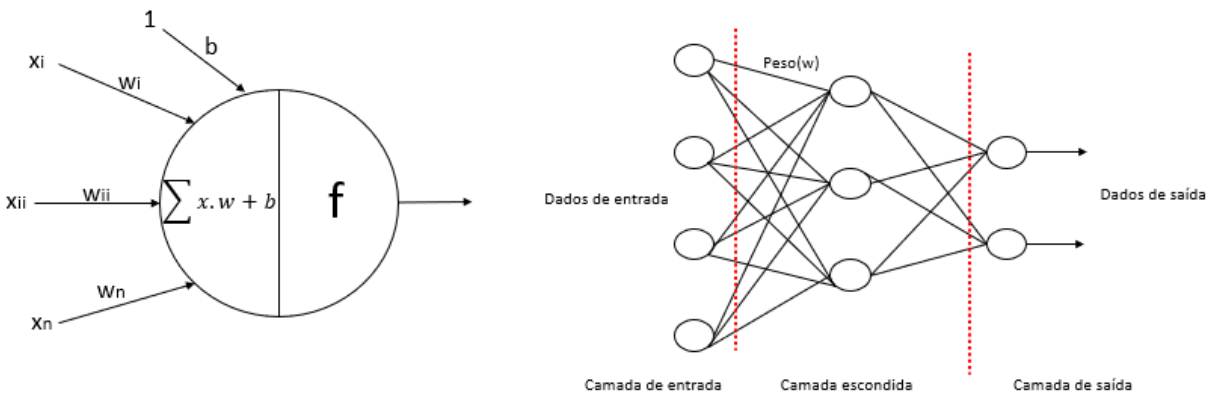
Figura 12 - Redes neurais biológicas e artificiais



Fonte: Elaboração própria (adaptado de Furtado, 2019)

A camada com os neurônios que recebem a informação externa é chamada de camada de entrada e a que os neurônios passam as informações processadas finais é a camada de saída. Os neurônios que não são de entrada ou saída são chamados de intermediários e as camadas, são chamadas de camadas escondidas (FURTADO, 2019). Cada camada é um algoritmo simples que vai processar a informação e a informação de saída de uma camada serve como informação de entrada para outra camada e segue assim até chegar na camada de saída.

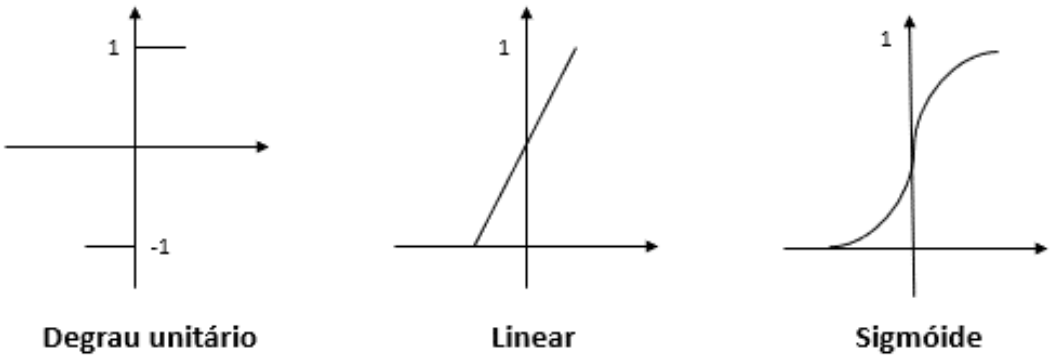
Figura 13 - Processamento de dados nas Redes Neurais Artificiais



Fonte: Elaboração própria(adaptado de Azarpour, 2017)

A Figura 13 representa o processamento nesses neurônios. Os dados de entrada(x) são recebidos pelos neurônios da camada de entrada, já associados com um peso(w), simulando as sinapses dos neurônios. Nessa camada é feita a soma ponderada dos dados com os respectivos pesos e adicionado a essa somatória o *bias*, uma constante de entrada com valor igual a 1 que é considerado um pseudo dado de entrada para contornar situações que os valores do padrão de entrada são iguais a zero. A função f é a função de ativação que é selecionada pelo usuário para cada modelo e que limita a amplitude do sinal de saída do neurônio, sendo as mais comuns degrau unitário, sigmóide unitário, linear demonstradas na Figura 14.

Figura 14 - Funções de ativação



Fonte: Elaboração própria

Alguns fatores que influenciam na precisão da previsão é a escolha do algoritmo de treinamento de Redes Neurais Artificiais que vai atualizando os valores dos pesos e *bias* de acordo com o método. Outra configuração a ser ajustada são quantas camadas escondidas serão utilizadas e quantos neurônios devem ter em cada camada.

2.3.1 Aplicação das redes neurais artificiais para previsão de taxa de desativação de catalisador

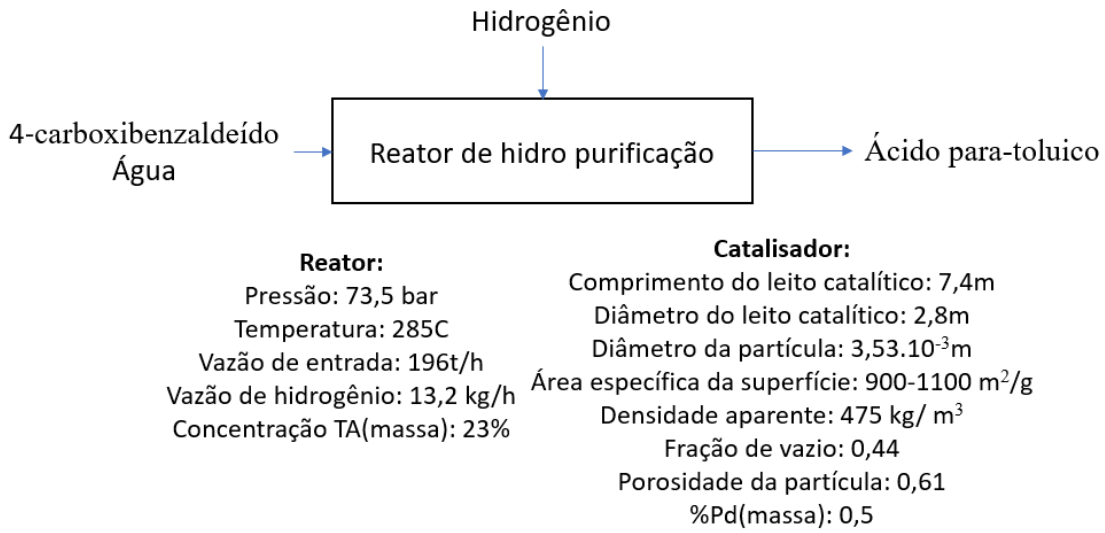
Com essa nova possibilidade do aprendizado de máquina, principalmente com as Redes Neurais Artificiais, Azarpour (2017) fez um estudo sobre desenvolver um modelo para prever o mecanismo de desativação de um catalisador em um processo químico, que é um fator muito importante para manter a alta produtividade e qualidade do produto. A desativação do catalisador pode ocorrer por conta de interações químicas, físicas, térmicas ou mecânicas, podendo ser por envenenamentos, incrustação, degradação térmica, formação de vapor, reações vapor-sólido ou sólido-sólido, atrito ou esmagamento com outros elementos.

O desafio de prever a desativação do catalisador se dá pois há vários mecanismos e suas diversas combinações para a desativação, principalmente considerando a área de superfície que muda conforme o tempo e torna o estudo mais complexo, sendo necessário um estudo laboratorial, com equipamentos de alto custo para analisar e simular as condições reais do reator. Aplicando Redes Neurais Artificiais, surge um novo método para tentar prever o valor da taxa de desativação do catalisador.

No estudo de Azarpour (2017), o objetivo é encontrar os parâmetros que influenciam no processo catalítico e foi aplicado um modelo híbrido que incorpora uma parte do processo com modelos matemáticos e a parte mais desafiadora do processo, que é prever a desativação do catalisador, é incorporado com redes neurais artificiais. Dado o objetivo do trabalho ser demonstrar aplicações de aprendizado de máquina na Engenharia Química, será mencionado somente a parte de redes neurais artificiais.

Para o estudo de Azarpour (2017), o algoritmo foi aplicado para dois estudos de caso. O primeiro caso é do reator de hidro purificação de ácido tereftálico bruto, em que 4-carboxibenzaldeído(4-CBA) é convertido em ácido para-toluico através de uma reação de hidrogenação com catalisador paládio com carbono que interfere tanto na pureza do produto, quanto na eficiência do processo de cristalização, centrifugação, filtração e secagem que procede a reação e produz o ácido tereftálico purificado. As condições de operação no reator de hidro purificação onde ocorre o processo catalítico, bem como o fluxograma dessa reação estão apresentadas na Figura 15.

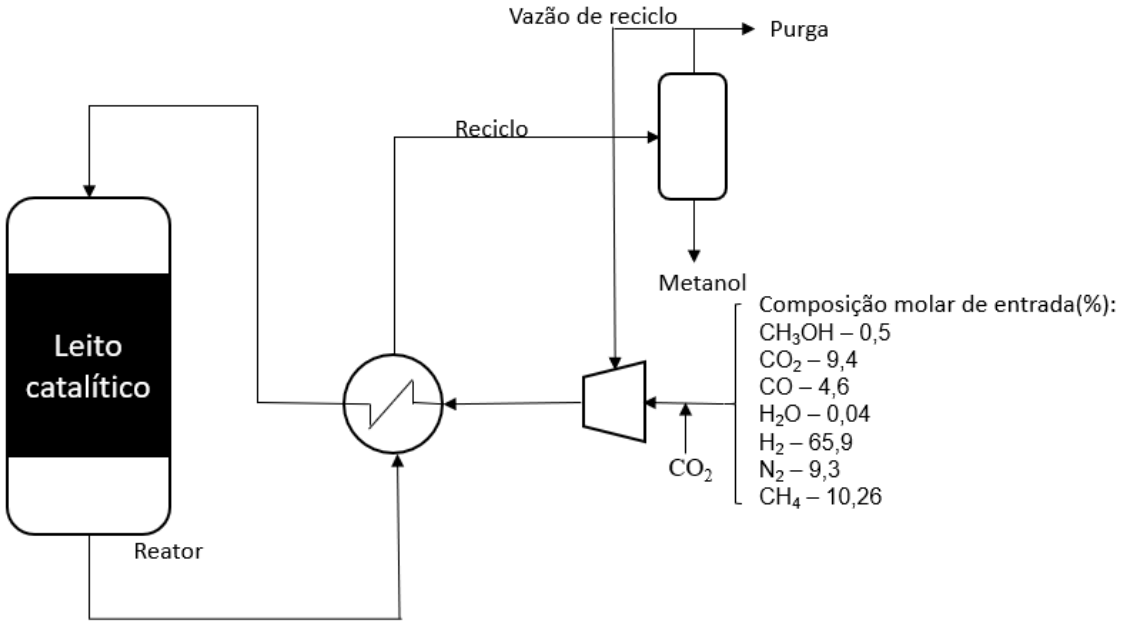
Figura 15 - Fluxograma da hidro purificação de ácido tereftálico bruto



Fonte: Elaboração própria (adaptado de Azarpour, 2017)

O segundo caso reportado por Azarpour (2017) é com a produção de metanol, em que o catalisador usado é $CuO\text{-}ZnO\text{-}Al_2O_3$. O metanol é produzido através da passagem de gases de síntese, incluindo hidrogênio, óxido de carbono e quaisquer gases inertes a uma temperatura e pressão elevadas ao longo de um ou mais leitos de catalisador. A alta conversão do metanol é dada geralmente a alta pressão e baixa temperatura e a desativação do catalisador interfere na taxa de produção do metanol. O metanol é geralmente recuperado por resfriamento do fluxo de gás do produto e separando-o do produto como um líquido e o gás não reagido é reciclado para o reator. As condições de operação, bem como o fluxograma dessa reação estão apresentadas na Figura 16.

Figura 16 - Fluxograma da produção de metanol



Reator:
Pressão: 76,98 bar
Temperatura de entrada: 230 °C
Temperatura do casco: 250 °C
Vazão molar de entrada por tubo: 0,6 mol/s
Número de tubos: 2962
Comprimento do tubo: 7,022 m
Diâmetro interno do tubo: 44,5 mm
Diâmetro externo do tubo: 48,5 mm
Condutividade térmica da parede: 48 W/m.K

Catalisador:
Diâmetro da partícula: 5.47×10^{-3} m
Densidade: 1770 kg/m^3
Capacidade calorífica: 5,0 kJ/kg.K
Condutividade térmica: 0,004 W/m.K
Fração de vazio: 0,4
Porosidade da partícula: 0,123

Fonte: Elaboração própria (adaptado de Azarpour, 2017)

Para a análise, foi considerado que os dados de entrada do algoritmo são os dados de concentração de entrada e saída do reator de cada componente no processo cada vez que a amostra é coletada para prever a taxa de reação e desativação do catalisador. Foi utilizado 70% dos dados para a etapa de treinamento e 30% para validação e teste. Para avaliar o erro do algoritmo, são utilizados balanços de massa e energia para calcular a taxa de reação e desativação do catalisador usando os dados de processo disponíveis e as especificações do catalisador.

Para cada estudo foi aplicado diferentes algoritmos (Levenberg-Marquardt, BFGS Quasi-Newton, Variable Learning Rate Gradient Descent, Gradient Descent with

Momentum) para serem empregados com diversas quantidades de neurônios na camada escondida. Para os dois casos foi utilizado somente uma camada escondida, que geralmente traz resultados satisfatórios. Foi estabelecido para ambos os casos que o objetivo seria obter um erro menor do que 10^{-18}. Após aplicado os diferentes algoritmos, foi comparado com o resultado esperado e calculado o erro. Na Tabela 1, está apresentado o erro para cada algoritmo e quantos neurônios foram empregados na camada escondida para cada caso.

Tabela 1 - Erro dos algoritmos para cada caso

Caso	Algoritmo de treinamento	Erro	Neurônios
Hidro purificação de ácido tereftálico	Levenberg-Marquardt	$1{,}1 \times 10^{-27}$	38
	Bayesian Regularization	$1{,}6 \times 10^{-5}$	23
	BFGS Quasi-Newton	$8{,}7 \times 10^{-5}$	26
	Resilient Backpropagation	$7{,}0 \times 10^{-2}$	15
Produção de metanol	Levenberg-Marquardt	$1{,}3 \times 10^{-19}$	31
	Bayesian Regularization	$1{,}1 \times 10^{-7}$	26
	BFGS Quasi-Newton	$2{,}9 \times 10^{-2}$	19
	Resilient Backpropagation	$3{,}8 \times 10^{-2}$	27

Fonte: Elaboração própria(adaptado de Azarpour, 2017)

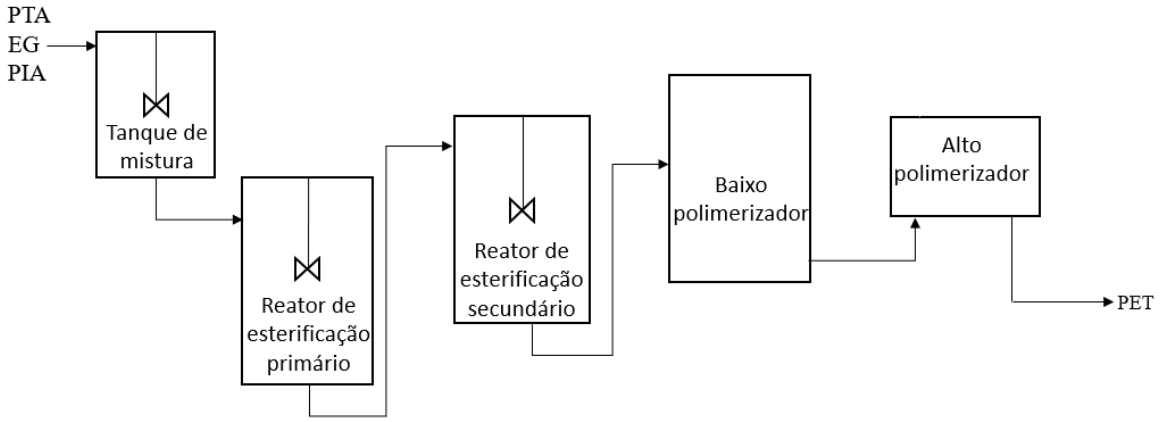
Em ambos os casos, o algoritmo de Levenberg-Marquardt foi o algoritmo que obteve o melhor resultado e o único em que o erro foi menor do que 10^{-18}. Assim, foi possível realizar a previsão da taxa de desativação do catalisador com o algoritmo de Levenberg-Marquardt com Redes Neurais Artificiais e evitar os cálculos complexos e também economizar tempo de elaboração do modelo.

2.3.2 Aplicação de soft sensor (sensor virtual) no controle de processos químicos

Outro estudo feito com aprendizado de máquina foi na área de controle de processos químicos. Os sensores verificam a diferença entre o valor desejado e o valor real de uma variável e o valor do desvio é enviado para um controlador, que ajusta o sistema para que a variável apresente o valor desejável. Porém, em algumas situações, a variável pode não ser facilmente medida ou precisa de um hardware específico de custo elevado para o controle aprimorado do processo (CARTWRIGHT, 2020). O aprendizado de máquina traz uma alternativa para que seja contornado essa situação como apresentado no estudo de Gonzaga (2009) em um processo de polimerização para a produção do Polietileno tereftalato (PET).

O estudo mostra o processo que ocorre em duas reações químicas, uma de esterificação e uma de polimerização. A resina de polietileno tereftalato é produzida através de uma pasta de ácido tereftálico purificado (PTA), etilenoglicol (EG) e ácido isoftálico purificado (PIA) que são misturados em um tanque de mistura. Essa pasta é introduzida em um reator de esterificação primário que produz os monômeros. Os monômeros são transferidos para o reator de esterificação secundário, em que a reação ocorre sob condições específicas e controladas de tempo de residência, temperatura e pressão para aumentar a taxa de esterificação. Em seguida, a polimerização ocorre também sob condições específicas de tempo de residência, temperatura e pressão em um baixo polimerizador, em que ocorre a reação para formar polímero com massa molecular da ordem de 2500 g/mol e EG como principal subproduto. O polímero segue para um alto polimerizador, que continua a polimerização até que a massa molecular alcance um valor de 15.000-20.000 g/mol, sendo muito importante o controle da massa molecular do polímero. O processo descrito está representado na Figura 17.

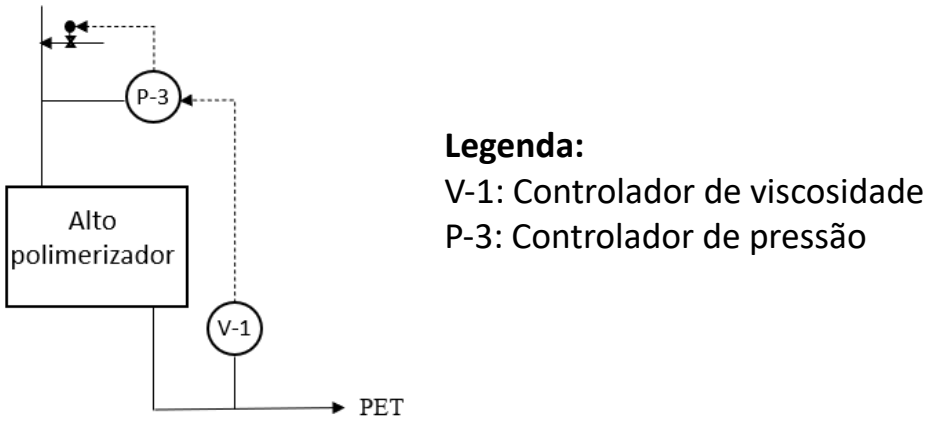
Figura 17 - Fluxograma da produção de PET



Fonte: Elaboração própria (adaptado de Gonzaga, 2009)

A viscosidade intrínseca é uma variável que pode ser diretamente medida por um viscosímetro e é uma medida relativa da massa molecular da resina, ou seja, é possível relacionar a massa molecular com a viscosidade intrínseca. Assim, o controle da massa molecular é feito na verdade com o controle da viscosidade intrínseca. O sinal do viscosímetro é enviado para o controlador de viscosidade que, em uma configuração de controle em cascata, define a pressão a ser fornecida pelo controlador no alto polimerizador. O controle da viscosidade é representado na Figura 18.

Figura 18 - Controle da viscosidade



Fonte: Elaboração própria (adaptado de Gonzaga, 2009)

Dessa forma, para a produção de Polietileno tereftalato (PET), uma especificação muito importante para a qualidade do produto é a viscosidade intrínseca, sendo a principal variável controlada do processo. Apesar de poder ser medida pelo viscosímetro, a temperatura do bloco do viscosímetro e o fluxo do polímero não desejáveis podem causar alguns problemas como a obstrução capilar e mau funcionamento da bomba e o viscosímetro pode acabar apontando um valor incorreto, que consequentemente afeta na qualidade da produção.

No estudo foi utilizado sensor virtual, um software em que várias medições são processadas juntas, juntamente com as redes neurais artificiais para prever o valor da viscosidade. A viscosidade intrínseca estimada é então utilizada para o controle desse processo. Geralmente, para uma Rede Neural Artificial, quanto maior a dimensão da entrada, maior deve ser o número de parâmetros, dados fornecidos e tempo necessário para a etapa de treinamento. Para contornar essa situação, o estudo selecionou um conjunto preliminar de oito variáveis do processo para a camada de entrada do modelo neural, considerando o conhecimento prévio. As variáveis selecionadas são a temperatura do reator de esterificação primário e secundário, segundo estágio do baixo polimerizador e o alto polimerizador; a pressão do primeiro e segundo estágio do baixo polimerizador e do alto polimerizador e, por fim, a vazão no baixo polimerizador. O padrão usado na camada de saída foi a viscosidade real obtida da planta pelo viscosímetro.

Foram coletados dados durante dois meses de operação para a etapa de treinamento e para avaliar o desempenho do modelo, tomando cuidado para selecionar somente os dados significativos com intervalo de confiança como critério de seleção. Para a Rede Neural Artificial, foi selecionada uma camada oculta usando tangente hiperbólica como funções de ativação. Cada conjunto foi testado pelo menos três vezes usando diferentes pesos e o número de neurônios na camada oculta. O modelo selecionado foi o que apresentou menor erro na fase de treinamento.

Foi comparado os dados previstos com os valores reais e o modelo resultou em uma boa performance em que os valores previstos se aproximaram dos valores reais durante um período de 40 horas de produção. O maior erro observado neste intervalo foi de 4 poise, que corresponde a um erro relativo de aproximadamente 0,3%. O valor previsto é enviado ao

sistema de controle do processo, que utiliza essa informação para manter a viscosidade em um valor pré-determinado na operação (GONZAGA, 2009).

2.3.3 Aplicações atuais na indústria

Por fim, apesar de não ser amplamente aplicado na indústria, o aprendizado de máquina já está presente em alguns processos de grandes empresas. Segundo Antaranews (2016), a Mitsui Chemicals, empresa japonesa que produz matéria-prima para fertilizantes químicos, junto com a NTT Communications, empresa de consultoria para soluções de tecnologia de informação para empresas, desenvolveram uma técnica para prever a qualidade do produto em um processo de produção de gás. Este processo é influenciado por vários fatores como temperatura, pressão e vazão. O modelo verifica a relação entre os dados das matérias-primas utilizadas na entrada do reator e as condições do reator com a concentração do gás-produto utilizando aprendizado de máquina. Isso resultou em um modelo com alta precisão, com uma diferença de no máximo 3% na concentração do gás do valor real.

A vantagem da previsão da qualidade do produto é que os operadores podem detectar sensores ou instrumentos de medição com defeitos, avaliar com precisão a condição da planta e verificar qualquer anomalia ao comparar os dados coletados e os valores previstos. Além disso, a empresa Mitsui Chemicals está estudando a partir desse caso uma forma de uma manutenção inteligente de plantas, o que pode resultar em uma produção mais segura e estável.

Outra aplicação de aprendizado na máquina que já está sendo utilizado na indústria, de acordo com Ottewell (2020), é o caso da Nouryon Industrial Chemicals, que em parceria com a Semiotic Labs, utilizou a tecnologia de autoaprendizado do SAM4 para prever quando manter ou substituir bombas e outros equipamentos, analisando a forma de onda da tensão e da corrente. Aplicando o SAM4 no processo produtivo, foram identificados alguns problemas na planta. Haviam depósitos de sal que caíam do transportador que aumentavam o atrito na bomba, identificando um problema também no processo de transporte. Também foi encontrada uma bomba superdimensionada, que consumia mais energia e tinha uma vida útil menor do que uma bomba dimensionada corretamente. Por fim, foi identificado uma

bomba em um local suscetível à cavitação. Com o uso do SAM4, a empresa conseguiu reduzir o consumo de energia das bombas em cerca de 15%.

Além de algumas empresas começarem a aplicar o aprendizado de máquina nos seus processos, seu potencial também chamou a atenção da Camille and Henry Dreyfus Foundation, uma instituição que promove a ciência química e Engenharia Química . Essa instituição possui um programa que fornece financiamento para projetos inovadores em qualquer área de aprendizado de máquina que tenha o objetivo do avanço das ciências químicas e a Engenharia Química. A Fundação prevê que esses projetos contribuirão com uma nova visão química fundamental e inovação no campo (DREYFUS, c2021).

3 PERSPECTIVA DO APRENDIZADO DE MÁQUINA NO ENSINO DE ENGENHARIA QUÍMICA

A aplicação do aprendizado de máquina na Engenharia Química ainda não é generalizada, com grande parte dos casos ainda em fase de pesquisa. Isso se deve ao fato de que para a implementação do aprendizado de máquina ser bem sucedida é necessário um grande volume de dados e a Indústria 4.0, grande responsável pela rápida expansão da quantidade de dados coletada dentro da indústria, ainda é um cenário recente.

Apesar disso, é provável que a aplicação de aprendizado de máquina na indústria continue a evoluir consideravelmente nos próximos anos. Os exemplos citados no capítulo anterior mostram o potencial que existe da aplicação dessa técnica na indústria química. Além disso, a Aspen Tech, empresa fornecedora de software e serviços amplamente utilizados nas indústrias de processo, adicionou em seu portfólio aplicações de IA e aprendizado de máquina para acompanhar a transformação digital na indústria (ASPENTECH, c2021).

Dado que a adoção de aprendizado de máquinas na indústria química é provável de se intensificar nos próximos anos, é importante preparar os alunos de Engenharia Química para esse cenário. O foco deve ser na familiarização dos conceitos básicos de aprendizado de máquina para que possam adequadamente identificar e avaliar oportunidades de aplicação em suas vidas profissionais.

Assim, será sugerido algumas oportunidades de inclusão do tema na grade curricular do estudante de Engenharia Química da Universidade Federal de São Carlos, com foco nas disciplinas apresentadas na Tabela 2.

Tabela 2 - Disciplinas para sugestão de modificação na grade curricular

Código	Nome da Disciplina	Créd	Semestre
10518-0	Projetos de Algoritmos e Programação Computacional para Engenharia Química	04	4
NA	Optativa Técnica – Introdução ao aprendizado de máquina na Engenharia Química	04	10

Fonte: Elaboração própria (adaptado de Curso de Graduação em Engenharia Química - Projeto Pedagógico)

Um ponto fundamental para o aprendizado de máquina é o desenvolvimento do algoritmo em uma linguagem de programação. Além de auxiliar no aprendizado de máquina, a programação pode ajudar a aprimorar soluções para problemas em geral. Durante o curso de Engenharia Química, os alunos realizam uma disciplina de Projetos de Algoritmos e Programação Computacional para Engenharia Química, que aborda estrutura algorítmica e linguagens de programação para serem utilizadas como ferramentas computacionais (Projeto Pedagógico). A disciplina é introduzida com planilhas eletrônicas e como utilizar suas ferramentas, indo de formatação condicional até macros. Em seguida, é ensinado sobre algoritmo, linguagens e estruturas de programação com a linguagem Visual Basic for Applications (VBA), que permite que se aplique uma linguagem de programação dentro das planilhas eletrônicas.

Nesta disciplina, o objetivo não seria aprofundar no aprendizado de máquina, mas sim introduzir a linguagem de programação Python devido ao seu destaque na área. A sugestão para esta disciplina seria substituir a linguagem Python no lugar de VBA por ser uma linguagem que está em grande uso devido ao fato de ter uma ampla gama de aplicações. Python consegue executar as mesmas tarefas que VBA e outras que não seriam possíveis de executar com o uso de bibliotecas que podem auxiliar em tarefas mais complexas e também consegue lidar com um volume maior de dados e executar de forma mais rápida. Para exemplificar, como VBA está vinculada com as planilhas eletrônicas, o número de dados máximo permitido fica limitado às linhas da planilha.

A linguagem Python também possui a possibilidade da integração com outras plataformas, inclusive com as próprias planilhas eletrônicas. Por fim, a linguagem Python é mais sucinta e, portanto, a compreensão e a escrita do código são feitas de forma mais direta e de mais fácil aprendizado. Isso é um ponto importante, principalmente pelo fato de a disciplina ser um dos primeiros contatos dos alunos com a programação computacional e contribuir com uma melhor familiaridade com a lógica de programação. Além da introdução da linguagem Python, também é interessante trazer uma discussão sobre as possibilidades da programação em Python, incluindo o aprendizado de máquina para iniciar o contato dos alunos com o assunto.

Outra sugestão para o curso é a criação de uma disciplina optativa técnica, Introdução ao aprendizado de máquina na Engenharia Química, que se aprofunda no aprendizado de máquina para os alunos que tiveram interesse na área. Para essa disciplina, o objetivo seria introduzir o aprendizado de máquina, aprofundar um pouco mais na linguagem Python com as bibliotecas, teoria e aplicações computacionais de alguns algoritmos de aprendizado de máquina e por fim aplicação do aprendizado de máquina na Engenharia Química.

Por ser uma disciplina que é desenvolvida através do computador e boa parte do conhecimento é adquirido na prática, é sugerido realizá-la de forma remota, com parte da carga horária dedicada a uma breve videoaula semanal para introduzir conceitos, fundamentos e exemplos. No final, apresentar exercícios que contemplem o conteúdo da aula para os alunos resolverem para entregar na semana seguinte como forma de avaliação da disciplina. A outra parte da carga horária seria dedicada a uma aula à distância em um horário específico, alguns dias após a videoaula ser disponibilizada, junto com o professor da disciplina para discutirem e tirar dúvidas sobre o assunto e os exercícios. Essa abordagem exige que haja maior participação dos alunos, visto que o maior aprendizado virá da resolução dos exercícios, obtendo o conhecimento na prática enquanto encontram as soluções por si mesmo.

Uma proposta de ementa com alguns tópicos interessantes a serem abordados estão descritos com mais detalhes a seguir.

Disciplina Optativa Técnica: **Introdução ao aprendizado de máquina na Engenharia Química**

Formato: Remoto

Conteúdo

1. O que é aprendizado de máquina?

Primeiramente, uma introdução sobre o que é aprendizado de máquina, suas aplicações, tipos de aprendizado de máquina como aprendizado supervisionado, não supervisionado, por reforço. Dentro dos tipos de aprendizado, mencionar alguns tipos específicos de aprendizado supervisionado como regressão e não supervisionado como clusterização.

2. Python

Caso a linguagem seja ensinada na disciplina Projetos de Algoritmos e Programação Computacional para Engenharia Química, realizar uma rápida revisão sobre a estrutura da linguagem. Caso a disciplina Projetos de Algoritmos e Programação Computacional para Engenharia Química se mantenha com a linguagem VBA, para a optativa, trazer uma breve introdução sobre a linguagem com a sua estrutura, variáveis, condicionais e funções. Essa introdução deverá ocupar menos carga horária do que na disciplina de Projetos de Algoritmos e Programação Computacional para Engenharia Química por conta de os alunos já terem tido um primeiro contato com a lógica de programação e como dito anteriormente, a linguagem Python é mais simplificada, sendo menos complexa de aprender.

3. Bibliotecas de Python

Após a revisão ou introdução da linguagem Python, explorar as bibliotecas mais importantes para a construção de um algoritmo de aprendizado de máquina como Pandas, Matplotlib e Scikit-learn. Na parte de Pandas, além de como utilizar a biblioteca, introduzir conceitos de Big Data e banco de dados, pois são esses dados que serão utilizados para a parte do treinamento de aprendizado de máquina. A utilização da biblioteca Matplotlib seria para auxiliar na construção de gráficos e mostrar resultados mais visuais. E por fim, Scikit-learn que é a biblioteca que apresenta pacotes específicos para aprendizado de máquina.

4. Algoritmos de aprendizado de máquina

Construção de algoritmos com alguns modelos de aprendizado de máquina como regressão linear e não linear, clusterização, árvores de decisão, redes neurais que estão presentes na biblioteca Scikit-learn. Com o fornecimento de uma base de dados, realizar as etapas de treinamento, avaliação e aprimoramento de parâmetros para aplicar no algoritmo.

5. Aplicação de Engenharia Química com aprendizado de máquina

Aplicar um algoritmo de aprendizado de máquina em problemas da Engenharia Química para ilustrar a possibilidade de aplicação na área e trazer o assunto mais para a realidade do estudante.

Ferramentas:

Gerenciador de ambientes Anaconda

Metodologia de ensino:

Parte da carga horária será dedicada a uma breve videoaula que será disponibilizada semanalmente para introduzir conceitos, fundamentos, exemplos e exercícios que contemplem o conteúdo da aula para que os alunos resolvam e entreguem na semana seguinte, sendo esses exercícios a avaliação da disciplina. A outra parte da carga horária será dedicada a uma aula à distância em um horário específico, três ou quatro dias após a videoaula ser disponibilizada, junto ao professor da disciplina para discutirem e tirarem dúvidas sobre o assunto e os exercícios.

Bibliografia básica:

GÉRON, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. Alta Books, 2019.
HAYKIN, Simon. **Redes neurais: princípios e prática**. Bookman Editora, 2007.
REN, Jingzheng et al. **Applications of Artificial Intelligence in Process Systems Engineering.** 2021.
SARKAR, D.; Bali, R.; Sharma, T. **Practical Machine Learning with Python.** New York: Apress, 2018.
SILVEIRA, Guilherme; BULLOCK, Bennett. **Machine Learning: introdução a classificação**. Editora Casa do Código, 2017.

Durante o conteúdo de introduzir o aprendizado de máquina, Python, bibliotecas e algoritmos, o(a) professor(a) responsável pela disciplina poderá disponibilizar vídeo aulas e materiais relacionados para o(a) aluno(a) aprender e absorver o conhecimento, praticando com os exercícios propostos. Para a parte da disciplina de aplicar um algoritmo de aprendizado de máquina em um problema da Engenharia Química (item 5 da ementa proposta), a abordagem poderá ser implementada através do desenvolvimento de um projeto em grupo pelos alunos. Os horários da disciplina poderão ser utilizados para o desenvolvimento do projeto e terá disponível reuniões do grupo com o(a) professor(a) responsável pela disciplina para auxiliar no andamento do projeto.

Para o projeto, o(a) professor(a) responsável pela disciplina poderá fazer a escolha de um único processo químico para todos os grupos trabalharem. Para obter os dados desse processo, cada grupo gera esses dados a partir de diversas simulações do processo químico com auxílio de um software de simulação de processo, como por exemplo, o Aspen Plus. Então, os dados gerados por todos os grupos serão unificados e compartilhados com todos os grupos e assim cada grupo trabalharia com o mesmo conjunto de dados. Desta forma, os(as) alunos(as) conseguem colaborar para a obtenção de um grande volume de dados, fator importante para que o modelo consiga realizar previsões precisas. Com os dados, cada grupo aplica o aprendizado de máquina com um tipo de algoritmo diferente e depois apresenta para a turma sobre a aplicação do aprendizado de máquina, falando um pouco mais sobre o algoritmo utilizado, processo do aprendizado de máquina, resultados e percepções.

Esses tópicos abordados trazem uma boa noção sobre o aprendizado de máquina para o estudante e podem despertar o interesse em estudar o assunto mais a fundo. Com essa maior disseminação sobre o aprendizado de máquina, aumenta-se o potencial de encontrar outras possíveis aplicações na Engenharia Química, auxiliando na otimização de processos.

Além disso, novas carreiras vêm surgindo para o engenheiro químico. Com o *Big Data*, a interpretação dos dados pode levar a maiores rendimentos e economia para e empresa. Essa nova tecnologia traz então um novo papel para o engenheiro químico. Os dados de fornecedores, chão de fábrica, vendas, marketing são oportunidades para o engenheiro químico analisá-los e trazer conhecimento para melhorar monitoramento e escolha de matérias-primas, acompanhamento do transporte e armazenamento de materiais em tempo real, a manutenção de equipamentos dentre outros (ABEQ, 2017). A inclusão do

aprendizado de máquina no currículo do aluno de Engenharia Química pode trazer um diferencial para essa área, dado que um dos métodos para a análise de dados é o aprendizado de máquina.

Em Abril de 2019, foi publicado as novas Diretrizes Curriculares Nacionais (DCNs) para os cursos de graduação em Engenharias, que abrange as competências esperadas do egresso de Engenharia. A proposta de inclusão do aprendizado de máquina no currículo da Engenharia Química está de acordo com as novas Diretrizes Curriculares Nacionais pois acredita-se que algumas competências esperadas possam ser trabalhadas como as seguintes, descritas no Art. 4º da Resolução Nº 2, de 24 de abril de 2019:

- Analisar e compreender os fenômenos físicos e químicos por meio de modelos simbólicos, físicos e outros, verificados e validados por experimentação:
    - ser capaz de modelar os fenômenos, os sistemas físicos e químicos, utilizando as ferramentas matemáticas, estatísticas, computacionais e de simulação, entre outras
    - prever os resultados dos sistemas por meio dos modelos
    - conceber experimentos que gerem resultados reais para o comportamento dos fenômenos e sistemas em estudo
    - verificar e validar os modelos por meio de técnicas adequadas
- Aprender de forma autônoma e lidar com situações e contextos complexos, atualizando-se em relação aos avanços da ciência, da tecnologia e aos desafios da inovação:
    - ser capaz de assumir atitude investigativa e autônoma, com vistas à aprendizagem contínua, à produção de novos conhecimentos e ao desenvolvimento de novas tecnologias.

Além das novas Diretrizes Curriculares Nacionais, a inclusão do aprendizado de máquina no ensino de Engenharia Química também vai de acordo com o que outras universidades pelo mundo estão fazendo, com iniciativas para introduzir o aprendizado de máquina na grade curricular. Essas universidades possuem disciplinas relacionadas ao aprendizado de máquina em seu programa para preparar os alunos para essa área que está

crescendo.

Um exemplo é que a Universidade Stanford possui em seu programa do curso de Bacharelado em Ciências em Engenharia Química uma disciplina “Abordagens de Ciência de Dados e Aprendizado de Máquina em Engenharia Química e de Materiais”. Essa disciplina aborda ciência de dados, estatística e aprendizado de máquina para desenvolver e aplicar métodos de aprendizado de máquina a uma série de problemas de engenharia com polímeros condutores, membranas de purificação de água, materiais de bateria, síntese orgânica e controle de qualidade na fabricação (STANFORD, c2020). A disciplina é optativa aos alunos do curso e uma boa oportunidade de os alunos terem maior conhecimento nessa área.

Outra universidade que aborda o aprendizado de máquina em seu curso é a Universidade de Toronto, que fornece a oportunidade para os alunos de graduação do curso de Engenharia Química de realizar um curso de Inteligência Artificial para complementar o programa acadêmico. O aluno poderá realizar um curso de três ou seis disciplinas, que abordam introdução à Inteligência Artificial e maior aprofundamento com análise de dados e aprendizado de máquina (UOFT, 2020). O curso é uma forma interessante do aluno ter o conhecimento e se aprofundar no assunto.

4 CONCLUSÃO

O aprendizado de máquina é um subconjunto da Inteligência Artificial que é capaz de analisar uma grande quantidade de dados para verificar padrões e fazer predições. Pode contornar algumas situações que necessitam de um grande esforço, tempo ou custo pois sua análise relaciona os dados de entrada com os dados de saída sem a necessidade de saber o modelo matemático por trás disso. Em contramão, é necessário uma grande quantidade de dados para que haja o treinamento e validação para que a máquina consiga aprender. Vem sendo amplamente utilizado para identificação de perfis de clientes, previsão de demanda e até mesmo na medicina, para diagnóstico de doenças. Na Engenharia Química, apesar de não ser amplamente aplicado, existem alguns estudos que apontam um grande potencial nas áreas de modelagem, controle e manutenção de processos.

Foi apresentado um estudo sobre como o aprendizado de máquina pode auxiliar para prever a taxa de desativação do catalisador de forma mais rápida do que por meio de modelos matemáticos. Também foi apresentado um estudo para o controle da viscosidade intrínseca na produção de um polímero, em que o uso do viscosímetro pode resultar em uma leitura incorreta. Porém, com uso de sensor virtual juntamente com as Redes Neurais Artificiais, é possível contornar essa situação e prever a viscosidade intrínseca. Algumas indústrias já utilizam o aprendizado de máquina nos seus processos, como exemplo da Nouryon Industrial Chemicals, que utiliza o aprendizado para prever a manutenção ou substituição de bombas do processo.

Com isso, é interessante que os alunos de Engenharia Química tenham mais contato com esse assunto. É interessante introduzir o aprendizado de máquina, sobre o que é, suas possibilidades e também conhecer as ferramentas que são usadas e como executá-las. A principal ferramenta é a linguagem de programação em que se constrói o modelo para o aprendizado de máquina. Uma linguagem que vem se popularizando na área é Python, que possui outras ferramentas bem desenvolvidas para apoiar na escrita e execução do código como bibliotecas e IDEs.

Assim, sugeriu-se algumas oportunidades de inclusão do tema na grade curricular do estudante de Engenharia Química da Universidade Federal de São Carlos para introduzir a linguagem de programação Python em uma disciplina de programação computacional para

familiarizar os alunos com a linguagem que é muito utilizada no aprendizado de máquina e incluir uma matéria optativa que se aprofunda um pouco mais no assunto, abordando mais sobre os fundamentos, ferramentas e aplicações.

5 REFERÊNCIA BIBLIOGRÁFICA

ABEQ. **Uma Engenharia Química 4.0**. Revista Brasileira de Engenharia Química. v. 33, n.1, p.7-10, 2017.

ANACONDA. 2021. Disponível em:<https://www.anaconda.com/>. Acesso em: 07/05/2021

ANGELONI, Maria. **Elementos intervenientes na tomada de decisão.** Ci. Inf., Brasília, 2003.

ANTARANEWS. **NTT Communications and Mitsui Chemicals succeeded quality prediction of chemical products in the production process using Artificial Intelligence (AI).** 15 de setembro de 2016. Disponível em: <https://en.antaranews.com/news/106717/ntt-communications-and-mitsui-chemicals-succeeded-quality-prediction-of-chemical-products-in-the-production-process-using-artificial-intelligence-ai> Acesso em: 26/05/2021

ASPENTECH. c2021. Disponível em:<https://www.aspentech.com/en>. Acesso em: 11/06/2021

AZARPOUR, Abbas et al. **A generic hybrid model development for process analysis of industrial fixed-bed catalytic reactors.** Chemical Engineering Research and Design, 2017.

BANARES-ALCANTARA, Rene; WESTERBERG, Arthur W.; RYCHENER, Michael D. **Development of an expert system for physical property predictions.** Computers & chemical engineering, 1985.

BHANDE, Anup. **What is underfitting and overfitting in machine learning and how to deal with it.** 11 de março de 2018. Disponível em: <https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76> Acesso em: 02/02/2021

BONACCORSO, G. **Machine learning algorithms.** Birmingham, Mumbai: Packt Publishing, 2017.

BROWNLEE, Jason. **What is a Hypothesis in Machine Learning?** Machine Learning Mastery. 4 de março de 2019. Disponível em:<https://machinelearningmastery.com/what-is-a-hypothesis-in-machine-learning/>. Acesso em 29/01/2021

CARTWRIGHT, Hugh M. **Machine Learning in Chemistry.** Royal Society of Chemistry, 2020.

COELHO, Pedro. **Rumo à Indústria 4.0.** Dissertação de Mestrado – FCTU Universidade de Coimbra, Portugal, 2016

DIÁRIO OFICIAL DA UNIÃO. **Resolução CNE/CES 2/2019.** Brasília, 26 de abril de 2019, Seção 1, pp. 43 e 44. Disponível em: <https://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=112681-rces002-19&category_slug=abril-2019-pdf&Itemid=30192>. Acesso em: 12/06/2021

DREYFUS. **Machine Learning in the Chemical Sciences and Engineering.** c2021. Disponível em: <https://www.dreyfus.org/machine-learning-in-the-chemical-sciences-and-engineering/>. Acesso em: 12/06/2021

FACULDADE UNYLEYA. **Saiba o que é a metodologia ativa e como aplicá-la.** Disponível em:<https://blog.unyleya.edu.br/inicie-sua-carreira/dicas-de-estudos1/saiba-o-que-e-a-metodologia-ativa-e-como-aplica-la/#Sala_de_aula_invertida> Acesso em: 13/06/2021

FURTADO, Maria Inês Vasconcellos. **Redes neurais artificiais: uma abordagem para sala de aula.** Atena Editora, 2019

GÉRON, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. Alta Books, 2019.

GOMES, Dennis dos Santos. **Inteligência Artificial: Conceitos e Aplicações.** Revista Olhar Científico – Faculdades Associadas de Ariquemes – V. 01, n.2, Ago./Dez. 2010

GONZAGA, J. C. B. et al. **ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process.** Computers & chemical engineering, v. 33, n. 1, p. 43-49, 2009.

GOTARDO, Reginaldo. **Linguagem de Programação I.** Rio de Janeiro: SESES, 2015.

HAJJAR, Zeinab; TAYYEBI, Shokoufe; EGHBAL AHMADI, M. H**. Application of AI in chemical engineering.** Artif Intell Emerg Trends Appl, 2018.

HASSABIS, Demis. **AlphaGo: using machine learning to master the ancient game of Go.** Google. 27 de janeiro de 2016. Disponível em: <https://blog.google/technology/ai/alphago-machine-learning-game-go/>. Acesso em: 28/01/2020

HAYKIN, Simon. **Redes neurais: princípios e prática**. Bookman Editora, 2007.

INTEL. **Saiba mais sobre Big Data: Medidas que Gerentes de TI Podem Tomar para Avançar com o Software Apache Hadoop.** 2013. Disponível em: <https://www.intel.com.br/content/dam/www/public/lar/br/pt/documents/articles/90318386-1-por.pdf>. Acesso em: 02/01/2021

IPYTHON. 2021. Disponível em: <https://ipython.org/install.html>. Acesso em 07/05/2021

JUPYTER. 2021. Disponível em:<https://jupyter.org/index.html>. Acesso em: 07/05/2021

MARSLAND, S. **Machine Learning: An Algorithmic Perspective.** 2 ed. Boca Raton:

Editora CRC Press Taylor & Francis Group, 2015.

MATPLOTLIB. Disponível em:<https://matplotlib.org/>. Acesso em: 07/05/2021

MEDINA, M.; Fertig, C. **Algoritmos e programação: teoria e prática.** São Paulo: Novatec; 2005

MENA, Isabela. **O que é a internet dos serviços.** Projeto Draft. 29 de agosto de 2018. Disponível em: <https://www.projetodraft.com/verbete-draft-o-que-e-internet-dos-servicos/>. Acesso em: 30/01/2021.

MENEZES, Nilo. **Introdução à Programação com Python.** 4 ed. São Paulo: Novatec Editora Ltda., 2014.

MOHRI, M.; Rostamizadeh A.; Talwalkar, A. **Foundations of Machine Learning.** 2ed. MIT Press, 2012.

NASTESKI, V. **An overview of the supervised machine learning methods.** HORIZONS.B, vol. 4, pp. 51–62, 12 2017.

NUMPY. 2021. Disponível em:<https://numpy.org/doc/stable/user/whatisnumpy.html>. Acesso em: 07/05/2021

NUNEZ, David; Borsato, Milton. **Panorama atual dos sistemas ciber-sísicos no contexto da manufatura.** Congresso Brasileiro de Gestão da Inovação e Desenvolvimento de Produtos - Itajubá, 2015.

OTTEWELL, Sean. **Machine Learning Glows Brighter.** Chemical Processing. 05 de agosto de 2020. Disponível em: < https://www.chemicalprocessing.com/articles/2020/machine-learning-glows-brighter/> Acesso em: 28/05/2021

PEREIRA, Adriano; Simonetto, Eugênio. **Indústria 4.0: conceitos e perspectivas para o Brasil.** Revista da Universidade Vale do Rio Verde, v. 16, n. 1, p.3-4 , jan./jul. 2018.

RASPBERRYPI. 2021. Disponível em: <https://www.raspberrypi.org/documentation/usage/python/>. Acesso em: 07/05/2021

REN, Jingzheng et al. **Applications of Artificial Intelligence in Process Systems Engineering.** 2021.

SACOMANO, José et al. **Indústria 4.0: conceitos e fundamentos.** 1 ed. São Paulo: Editora Edgard Blücher Ltda, 2018

SARKAR, D.; Bali, R.; Sharma, T. **Practical Machine Learning with Python.** New York: Apress, 2018.

SCKIT-LEAN. 2021. Disponível em:<https://scikit-learn.org/stable/getting_started.html >. Acesso em: 07/05/2021

SEABORN. 2021. Disponível em: <https://seaborn.pydata.org/>. Acesso em: 07/05/2021

SEBESTA, Robert. **Conceitos de Linguagem de Programação.** 9 ed. Porto Alegre: Bookman, 2011.

SILVEIRA, Guilherme; BULLOCK, Bennett. **Machine Learning: introdução a classificação**. Editora Casa do Código, 2017.

SPRINGBOARD. **Best language for Machine Learning: Which Programming Language to Learn.** India, 31 de agosto de 2020. Disponível em: <https://in.springboard.com/blog/best-language-for-machine-learning/>. Acesso em: 27/04/2021

STANFORD. **Bachelor of Science in Chemical Engineering.** Stanford Explore Degrees. c2020. Disponível em: <https://exploredegrees.stanford.edu/schoolofengineering/chemicalengineering/#bachelorstext> Acesso em: 29/06/2021

TRAN, Khuyen. **Top 6 Python Libraries for Visualization: Which one to use?** Towards Data Science. 24 de julho de 2020. Disponível em:<https://towardsdatascience.com/top-6-python-libraries-for-visualization-which-one-to-use-fe43381cd658>. Acesso em: 07/05/2021

UFSCAR. **Curso de Graduação em Engenharia Química - Projeto Pedagógico.** Universidade Federal de São Carlos. São Carlos. Disponível em:<http://www.prograd.ufscar.br/cursos/cursos-oferecidos-1/engenharia-quimica/engenharia-quimica-projeto-pedagogico.pdf> Acesso em: 18/05/2021

UOFT. **Minors & Certificates.** University of Toronto Engineering. c2020. Disponível em: <https://undergrad.engineering.utoronto.ca/academics-registration/minors-certificates/>. Acesso em: 29/06/2021

VENKATASUBRAMANIAN, Venkat. **The promise of artificial intelligence in chemical engineering: Is it here, finally?** AIChE Journal, v. 65, n. 2, 2019.

YEGULALP, Serdar. **Julia vs. Python: Which is best for data science?** InfoWorld, 27 de maio de 2020. Disponível em: <https://www.infoworld.com/article/3241107/julia-vs-python-which-is-best-for-data-science.html>. Acesso em 28/04/2021.

ZHANG, Alina. **Pandas, Dask or PySpark? What Should You Choose for Your Dataset?** Data Driven Investor. 21 de agosto de 2019. Disponível em:<https://medium.datadriveninvestor.com/pandas-dask-or-pyspark-what-should-you-choose-for-your-dataset-c0f67e1b1d36>. Acesso em: 07/05/2021