

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



CONTROLE DE ESTOQUE UTILIZANDO REDES NEURAIS

LUCAS DE SOUZA FERREIRA

GOIÂNIA

2021

LUCAS DE SOUZA FERREIRA

CONTROLE DE ESTOQUE UTILIZANDO REDES NEURAIIS

Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Gustavo Siqueira Vinhal

Banca examinadora:

Prof. Ma. Lucilia Gomes Ribeiro

Prof. Me. Fernando Gonçalves
Abadia

GOIÂNIA

2021

LUCAS DE SOUZA FERREIRA

CONTROLE DE ESTOQUE UTILIZANDO REDES NEURAIIS

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Ciência da Computação, em ____/____/____.

Orientador: Prof. Me. Gustavo Siqueira Vinhal

Prof. Ma. Lucilia Gomes Ribeiro

Prof. Me. Fernando Gonçalves Abadia

Prof^a. Ma. Ludmilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de
Curso

GOIÂNIA

2021

DEDICATÓRIA

Aos meus pais e amigos que foram os maiores apoiadores nessa trajetória.

AGRADECIMENTOS

Ao Professor Gustavo Siqueira Vinhal, orientador acadêmico, pelo apoio e paciência no decorrer do trabalho.

A todos que direta ou indiretamente colaboraram para materialização deste trabalho.

RESUMO

No presente trabalho foi utilizado redes neurais recorrentes para a previsão de vendas. A previsão de vendas no varejo desempenha um papel importante na cadeia de operação de uma empresa. Com base nessa previsão é possível negociar e planejar junto aos fornecedores a compra e entrega dos produtos necessários. O desenvolvimento de descoberta de conhecimento em base de dados é um processo amplo que abrange diversos conceitos e processos. A mineração de dados é uma das várias etapas do processo, que contribui para tomadas de decisões nas empresas. No presente trabalho foram utilizados modelos de classificação para minerar dados de vendas de produtos relacionados a uma empresa do ramo farmacêutico. Mais especificamente, foram abordados os fatores referentes aos dados comerciais de vendas, considerando um período aproximado de 10 anos. Após todo processamento de descoberta em base de dados e mineração de dados, foi utilizado as bibliotecas Pandas e Tensorflow da linguagem *python* para processar os dados da venda em uma rede neural de recorrência, a fim de prever a quantidade mensal de vendas.

Palavras-Chave: classificação, previsão de venda, mineração de dados, rede neural de recorrência.

ABSTRACT

In the present work, recurrent neural networks were used to forecast sales. Retail sales forecasting plays an important role in a company's chain of operation. Based on this forecast, it is possible to negotiate and plan with suppliers the purchase and delivery of the necessary products. The development of knowledge discovery in databases is a broad process that encompasses several concepts and processes. Data mining is one of several steps in the process that contributes to decision making in companies. In the present work, classification models were used to mine sales data for products related to a pharmaceutical company. More specifically, factors related to commercial sales data were addressed, considering an approximate period of 10 years. After all the discovery processing in the database and data mining, the Pandas and Tensorflow libraries of the python language were used to process the sales data in a recurrence neural network, to predict the monthly sales quantity.

Keywords: classification, sales forecasting, data mining, recurrence neural network.

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas de DCBD	13
Figura 2 – Biologia de uma célula nervosa e seu mecanismo de comunicação	15
Figura 3 – Relação entre uma rede neural biológica e artificial	15
Figura 4 – Modelo de um neurônio artificial – <i>Perceptron</i>	16
Figura 5 – Processo de recorrência em uma rede neural	18
Figura 6 – As diferentes aplicações de RNR	19
Figura 7 – Vendas diariamente	22
Figura 8 – Vendas mensalmente	22
Figura 9 – Função de ativação ELU, graficamente demonstrada	27
Figura 10 - Resultado experimental do treinamento da rede neural recorrente utilizando dados de vendas	30

LISTA DE TABELAS

Tabela 1 – Sequências para o treinamento.	27
Tabela 2 – Erro médio absoluto a cada 2000 iterações e erro médio absoluto total do teste	30

SUMÁRIO

1. INTRODUÇÃO	10
2. REFERENCIAL TEORICO	12
2.1. Descoberta de conhecimento em base de dados – DCBD	12
2.2. Mineração de dados.....	13
2.3. Redes Neurais Artificiais.....	14
2.3.1.Neurônios Biológicos e Neurônios Artificias	14
2.3.2.Principais arquiteturas de RNA.....	17
2.4. Redes Neurais Recorrentes (RNR)	18
2.4.1.Redes Elman e Jordan.....	20
3. MATERIAIS E MÉTODOS.....	22
3.1. Problema.....	22
3.2. Base de dados	22
3.3. <i>Hardware</i>	22
3.4. Caracterização e análise dos dados.....	22
3.5. Python.....	24
3.5.1.Bibliotecas	24
4. Desenvolvimento	26
4.1. Processamento do Algoritmo.....	26
5. RESULTADO	30
6. CONCLUSÃO	32
REFERÊNCIAS	33
APÊNDICE A – DADOS DE VENDAS MENSAS DE MEDICAMENTO ENTRE NOVEMBRO DE 2011 E JUNHO DE 2021	35

1. INTRODUÇÃO

A previsão de vendas no varejo desempenha um papel importante na cadeia de operação de uma empresa. Com base nessa previsão é possível negociar e planejar junto aos fornecedores a compra e entrega dos produtos necessários. As previsões devem levar em consideração diversas variáveis tais como a disponibilidade do produto perante o fornecedor, perfil de consumo do cliente, sazonalidade do produto, entre outros. Uma previsão bem-feita pode alavancar as vendas, enquanto uma malfeita pode afetar toda a empresa, ocasionando em prejuízos. Empresas estão constantemente coletando dados oriundos de suas transações comerciais. Analisar esses dados de forma inteligente, auxiliam os empresários a tomarem decisões de negócio de maneira mais assertiva para o seu negócio.

A mineração de dados ou *data mining*, utiliza ferramentas estatísticas para o tratamento de dados e técnicas quantitativas para extrair o máximo de conhecimento existente em uma base de dados. As técnicas de mineração de dados, incluem métodos de inteligência artificial (sistemas especialistas, lógica *fuzzy*, redes neurais, entre outros), árvores de decisão, métodos de indução de regras, redes neurais, entre outras (GARGANO & RAGGAD, 1999). Incluem também o uso de métodos estatísticos convencionais (HAND, 1998).

Ao utilizar as técnicas e ferramentas de mineração de dados nos dados armazenados de empresas, é possível detectar padrões e extrair conhecimento de forma inteligente e confiável. Essa extração de conhecimento é importante pois os dados refletem a vida da empresa. Eles carregam informações importante sobre as operações realizadas, perfis de clientes e muito mais. A tomada de decisão utilizando essas informações aumenta a probabilidade de sucesso da empresa.

A aplicação de técnicas de mineração de dados na análise dos requisitos de estoque de uma empresa pode evitar excesso ou falta de demanda, preparar melhor a equipe de vendas para atingir os objetivos especificados, auxiliar na formulação de ações de marketing e atrair investidores.

De acordo com levantamento da Confederação Nacional do Comércio (CNC), com mais de 6.000 lojas no Brasil, no início de junho de 2018 aproximadamente 30,7% dos estoques de bens duráveis nessas lojas estavam acima do nível adequado. Em termos de evolução do consumo, 14,3% das lojas ainda possuem estoque abaixo do ideal.

O estoque excessivo leva a fundos fixos no negócio. Isso significa custos adicionais de estrutura e manutenção, além de produtos perecíveis que podem vencer a qualquer momento. Portanto, eliminar ou evitar esse excedente é fundamental e elimina o tempo e o dinheiro desperdiçados pela empresa. O estoque insuficiente também é ruim. Ele implica em uma má reputação da empresa entre os clientes, porque as necessidades da empresa não foram resolvidas, e os clientes também pensam que isso é um tipo de negligência.

Diante do exposto, este trabalho propõe a utilização de mineração de dados na aplicação de um sistema de previsão de demanda de estoque. Para isso, ele está estruturado como segue. O Capítulo 2 apresenta o Referencial Teórico, explanando o problema e teoria acerca das técnicas utilizadas neste trabalho. O Capítulo 3 apresenta os materiais utilizados no trabalho. O Capítulo 4 apresenta os resultados e o 5 conclui o trabalho.

2. REFERENCIAL TEORICO

Para que uma empresa se mantenha competitiva em um mercado concorrido e globalizado, o seu gestor precisa ter em mãos dados confiáveis, para que ele tome as melhores decisões possíveis. Ter em mãos dados para previsão de vendas, demandas e estoques futuros, é um elemento chave para uma boa gestão. “É preciso haver um norte para que a administração da produção possa trabalhar. A previsão de vendas oferece este direcionamento” (PEINALDO, 2007).

Segundo (Ritzman, Krajewski, & Klassen, 2004), o planejamento eficaz da demanda do cliente é um dos principais fatores que predizem o sucesso da cadeia de abastecimento preciso. Dados mais recentes do (SEBRAE, 2014) apontam que 50% das empresas fecharam “as portas” no período do estudo (2014) não definiu uma estratégia para evitar o desperdício. Esse desperdício geralmente está relacionado ao planejamento deficiente de estoque.

2.1. Descoberta de conhecimento em base de dados – DCBD

Entre as grandes quantidades de dados que a empresa gera todos os dias, concentre-se a necessidade de dados estruturados para que possam ser extraídos conhecimento. As empresas dependem cada vez mais de dados, para melhorar a tomada de decisão (FEIJÓ, 2013).

O processo de descoberta de conhecimento do banco de dados (DCBD) é um processo interativo para identificar novos padrões eficazes nos dados e pode ser útil e explicável (FAYYAD; PIATETSKY-SHAPIRO; SMITH, 1996). Em um nível abstrato, o campo DCBD se concentra em métodos e técnicas de compreensão de dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Para implementar o processo DCBD, as metas devem ser claras, além do resultado a ser alcançado e as Regras de negócios e aplicativos. Segundo Figueira (1998), as etapas do DCBD podem ser divididas nas seguintes categorias:

- **Seleção:** esta é a etapa de agrupar (organizar) os dados;
- **Pré-processamento:** neste momento, os dados são ajustados. Ao final do processo, devem estar no formato correto, não podendo ser repetidos e possuir outras características;

- **Conversão:** esta é a etapa de armazenamento de dados para promover o uso da tecnologia de mineração de dados;
- **Mineração de dados:** é a principal atividade do conhecimento. Algoritmos de descoberta de padrões são aplicados para esse fim;
- **Interpretação e avaliação:** esta etapa envolve interpretar os dados gerados e verificar se eles são válidos para as perguntas feitas.

A Figura 1 mostra as etapas do DCBD ilustradas.

Figura 1 - Etapas de DCBD.



Fonte: (FAYYAD, 2001).

2.2. Mineração de dados

A mineração de dados é um ramo da computação que começou na década de 1980, quando profissionais em empresas e organizações estavam começando a se preocupar com a quantidade de dados do computador armazenados e não utilizados nas empresas. A mineração de dados essencialmente extrai informações de enormes bancos de dados da forma mais automatizada possível. Atualmente, a mineração de dados inclui, principalmente, análise dos dados extraídos. Assim, é possível procurar e investigar as necessidades reais de cada cliente e criar atividades de marketing específicas. Portanto, uma empresa de cartão de crédito, por exemplo, pode aplicar mineração de dados para extrair informações importantes do seu cliente (como gastos, hábitos de compra, nível de dívidas, entre outros) com o objetivo de oferecer serviços específicos.

A mineração de dados também pode ser definida com base no uso de tecnologia estatísticas ou analíticas para processar e analisar dados brutos para encontrar padrões e tendências importantes (CHIU; TAVELLA, 2006). A tarefa de mineração está dividida em 2 (dois) níveis (SILVA; MARQUES; Bocarioli

2006). Um deles é a tarefa de prever uma classe baseada na análise de atributos. O outro nível são as tarefas descritivas, que são projetadas para encontrar padrões que descrevam os dados de uma forma que os humanos consigam explicar.

Ao definir o escopo do problema, é necessário definir a tecnologia a ser implementada, como classificação, Regressão, associação ou agrupamento. Este trabalho visa a implementação de uma Rede Neural Artificial para mineração de dados.

2.3. Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são inspiradas no entendimento biológico do sistema nervoso humano (LIPPMANN, 1987). Em 1943 Warren McCulloch e Walter Pitts criaram modelos computacionais de redes neurais baseados em matemática e algoritmos chamados de lógica de limiar. Porém, só foram ganhar notoriedade algum tempo depois, quando foram introduzidos novos algoritmos de treinamento como, por exemplo, o algoritmo de *backpropagation* que em tradução livre fica “retro propagação” (após o treinamento, o algoritmo “volta” para aperfeiçoar o resultado).

Segundo Grübler (2018), essas são as principais aplicações das RNAs:

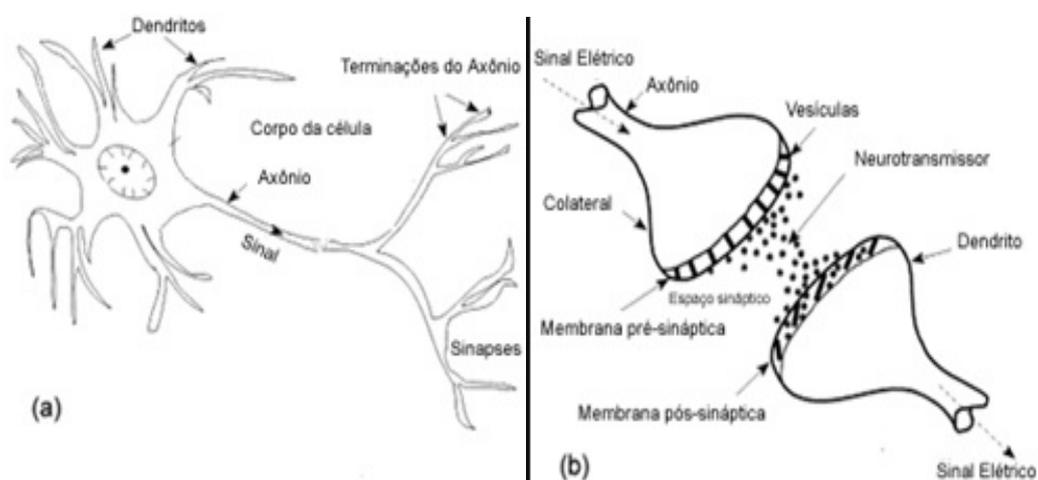
- Processamento de linguagem natural;
- Reconhecimento de fala e imagens;
- Previsão de valores.

2.3.1. Neurônios Biológicos e Neurônios Artificiais

O sistema nervoso biológico tem a estrutura mais complexa. Esses sistemas são compostos de células nervosas ou neurônios que desempenham diferentes funções. O corpo celular desse tipo de célula possui dois tipos de ramos: dendritos e axônios, que podem ser vistos na Figura 2 (a).

Além de plasma contendo substâncias moleculares necessárias para o funcionamento celular, o corpo celular também carrega informações importantes sobre suas propriedades. A comunicação entre os neurônios ocorre por meio de impulsos captados pelos dendritos, que são responsáveis por receber informações e transmiti-las ao corpo celular por meio dos axônios. Os axônios são divididos em ramos colaterais, que recebem sinais do corpo celular e os transmitem aos dendritos, que transmitem a outros dendritos de neurônios vizinhos por meio das sinapses, conforme mostrado na Figura 2 (b).

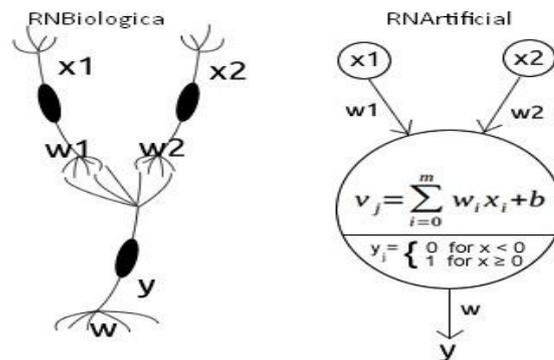
Figura 2 - Biologia de uma célula nervosa e seu mecanismo de comunicação.



Fonte: (Grübler, 2018).

A semelhança entre redes artificiais e biológicas se concentra nos axônios e dendritos e na forma de se comunicar por meio de sinapses. A Figura 3 mostra a representação dessa relação.

Figura 3 - Relação entre uma rede neural biológica e artificial.

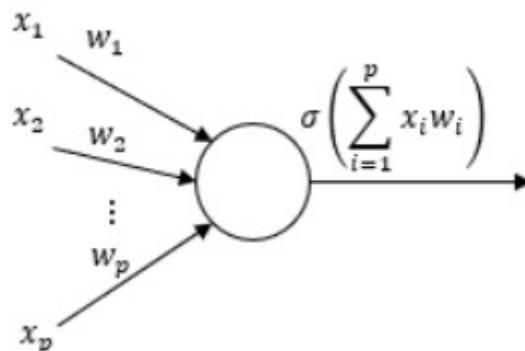


Fonte: (Grübler, 2018).

De acordo com a Figura 3, x representa o sinal recebido por outro neurônio e w é a intensidade da sinapse recebida (peso). Ambas as redes têm a aptidão de ajustar a amplitude das sinapses em uma série de camadas interconectadas. O modelo artificial mostrado na Figura 3 simula a rede neural artificial mais simples, chamada de *perceptron*.

O modelo multicamada *perceptron* é basicamente um modelo matemático que tem n entradas e apenas uma saída binária. Esse modelo foi proposto por Frank Rosenblatt, na década de 50 (ROSENBLATT, 1958). Esse modelo matemático foi inspirado nos trabalhos de Warren McCulloch e Walter Pitts. Hoje em dia quase não usa mais esse modelo devido a sua simplicidade, além de não obter resultados satisfatórios para problemas mais complexos. A Figura 4 representa um neurônio artificial a partir do modelo matemático proposto por Frank.

Figura 4 - Modelo de um neurônio artificial – Perceptron.



Fonte: (RAUBER, 2005).

De acordo com a Figura 4, cada entrada X_p tem um valor e um associado chamado de peso (W_p). A saída do *perceptron* é dada pela Equação 1.

$$y_j = \sigma\left(\sum_{i=1}^p w_i x_i - \mu\right) \quad (1)$$

De acordo com a Equação 1, um limiar μ é subtraído da combinação linear entre as entradas e seus respectivos pesos. Se o resultado for maior do que 0, a saída é positiva ($y = +1$). Caso contrário, a saída é negativa ($y = -1$) (VITORIA, 2020).

2.3.2. Principais arquiteturas de RNA

Quando se refere à uma arquitetura de RNA está falando sobre a disposição dos neurônios, em relação aos outros, seguindo suas conexões sinápticas. As principais arquiteturas são rede *feedforward*. Nessa arquitetura, a rede é progressiva e a saída de um neurônio é conectada a outro neurônio na próxima camada na direção esquerda / direita. Cada arquitetura tem sua necessidade de uso e as principais são:

- **Feedforward de camadas simples:** esse tipo de arquitetura os neurônios são dispostos de apenas uma camada. Seu principal exemplo são os *perceptron*;

- **FeedForward de Camadas Múltiplas:** essa arquitetura foi desenvolvida com intuito de lidar com problemas não linearmente separáveis, que são aqueles onde não se pode separar 2 classes distintas no eixo cartesiano bidimensional apenas traçando uma reta (LANHELLAS, 2018). Consiste em um grupo de neurônios chamados "nós". A rede possui uma camada de entrada (sem funções de computação), uma ou mais camadas ocultas e uma camada de saída. Exemplos: *Perceptron* Multicamadas, Redes de Base Radial.
- **Recorrente ou Realimentada:** sua saída é realimentada para outros neurônios como um sinal de entrada. Então é usado para processar sistemas variáveis no tempo. Eles são usados para previsão de séries temporais, otimização, identificação de sistema e controle de estoque. Exemplo: redes de Elman e Jordan.

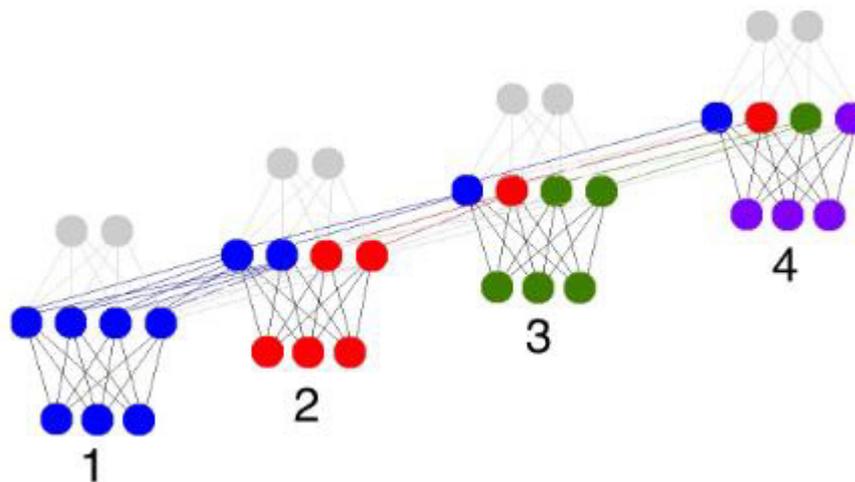
2.4. Redes Neurais Recorrentes (RNR)

Existem várias situações na natureza em que um determinado evento é uma consequência direta de eventos anteriores. Esses eventos são chamados de eventos sequenciais. Sob essas circunstâncias, os dados ou valores fornecidos não podem ser avaliados individualmente. Faz parte de uma sequência de outros valores. Existem alguns exemplos de eventos contínuos que podem ser observados todos os dias: quadros no vídeo, memorização de letras, palavras em frases coerentes, preços da bolsa de valores, temperatura, taxa de juros ou uma venda realizada em uma transação.

A rede neural *feedforward* pode mapear a informação de entrada de acordo com os pesos sinápticos conectados entre seus neurônios, processá-la e gerar saída. No entanto, ele não pode armazenar e processar informações no tempo, que é a característica básica para fazer previsões e garantir a memória de curto prazo. Se a informação passada de uma camada de neurônios para a próxima camada de neurônios também é passada para ela (isso é chamado de recorrência), esse recurso pode ser obtido. Portanto, o neurônio não apenas tem informações da entrada atual, mas também tem informações da entrada anterior.

Segundo (Haykins, 2008) “redes recorrentes são redes neurais com um ou mais laços de realimentação. A realimentação pode ser global ou local”. A Figura 5 mostra o processo de recorrência em uma rede neural.

Figura 5 processo de recorrência em uma rede neural.

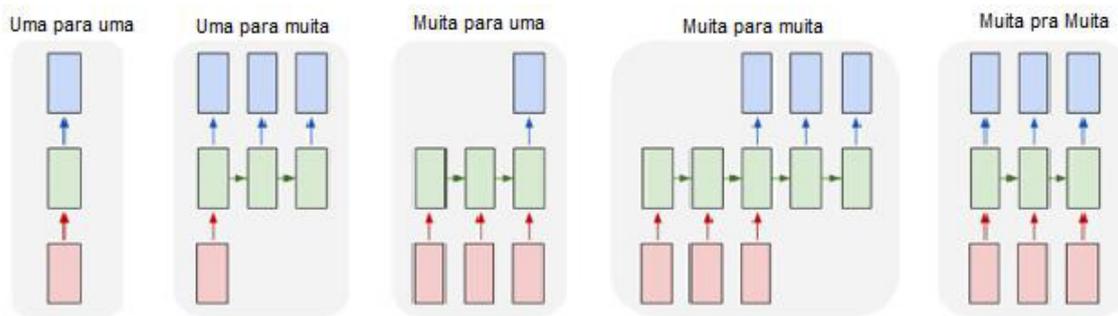


Fonte: (Trask, 2018)

Conforme ilustra a Figura 5, no primeiro caso, o estado oculto é definido puramente por dados, ou seja, definido pela camada de entrada. O segundo estado é uma mistura do primeiro estado oculto e os dados observados durante o segundo período. Além de conter informações dos dados observados no terceiro período, o terceiro estado oculto também contém uma mistura do primeiro e do segundo estados ocultos. Finalmente, o quarto estado oculto contém algumas informações sobre o que aconteceu em todos os períodos.

A Figura 6 mostra diferentes tipos de aplicações de RNR.

Figura 6 As diferentes aplicações de RNR.



Fonte Adaptado (Karpathy,2015)

De acordo com a Figura os tipos de aplicações de RNR são:

- **Uma para uma:** Que é a clássica rede neural;
- **Uma para muita:** Entrada apenas 1 dado, várias camadas ocultas e saída, essas camadas ocultas trabalha com recorrência, fazendo com que as informações anteriores sejam passadas a frente.
- **Muita para uma:** Os dados de entrada são passados diversas vezes, e processados na camada oculta, para enfim sair apenas um dado de saída.
- **Muita para Muita:** São lidas várias entradas em um período antes da camada oculta começar a processar a previsão. Fazendo com que esse tipo de rede neural tenha uma defasagem temporal entre a entrada e a saída da previsão.
- **Muita para Muita:** Representada pelo quarto tipo da Figura 6, ela é mais utilizada quando há uma necessidade de ter várias entradas e várias saídas no mesmo período.

2.4.1. Redes Elman e Jordan

A rede de Elman, proporciona uma unidade de contexto, que armazena as ativações anteriores e retorna camadas ocultas usando esses valores armazenado. A utilização de realimentação na camada oculta permite que todas as amostras já apresentadas à rede influenciem em sua resposta, enfatizando a influência sobre entradas mais recentes (Braga et al., 2007). Na rede Elman, existem uma camada de entrada, uma camada de saída e uma camada de contexto, cujo objetivo principal é lembrar a ativação da camada anterior, o que

pode ser considerado um atraso de uma etapa. Na rede Elman, essa camada de contexto estabelece conexões entre as camadas ocultas.

Semelhante à rede de Elman, na rede de Jordan, a saída é copiada para a unidade de contexto. A maior diferença entre as topologias de rede Jordan e Elman é que, nas redes Jordan, o loop vai da saída à entrada. Outra característica da rede Jordan é que o número de neurônios na unidade de contexto deve ser igual ao número de neurônios na camada de entrada.

3. MATERIAIS E MÉTODOS

Neste capítulo vai ser apresentados, os materiais e métodos que foram utilizados para um desenvolvimento do trabalho.

3.1. Problema

O presente trabalho tem como problema principal a previsão de estoque no varejo. O objetivo principal é aplicar RNA em um algoritmo capaz de realizar essa previsão. Para isso, serão utilizadas redes neurais recorrentes capazes de prever a partir dos dados históricos de vendas.

3.2. Base de dados

Os dados utilizados nesse trabalho foram cedidos por uma empresa sediada em Goiânia – GO. Tal empresa pertence ao ramo farmacêutico, do qual foram coletados dados entre 23 de novembro de 2011 até o dia 01 de junho de 2021

Ao todo foram coletados 35764 registros de vendas de um determinado produto de alto giro. Os dados estão separados em 3 colunas: ano da venda, mês da venda, e quantidade de venda em cartelas no mês;

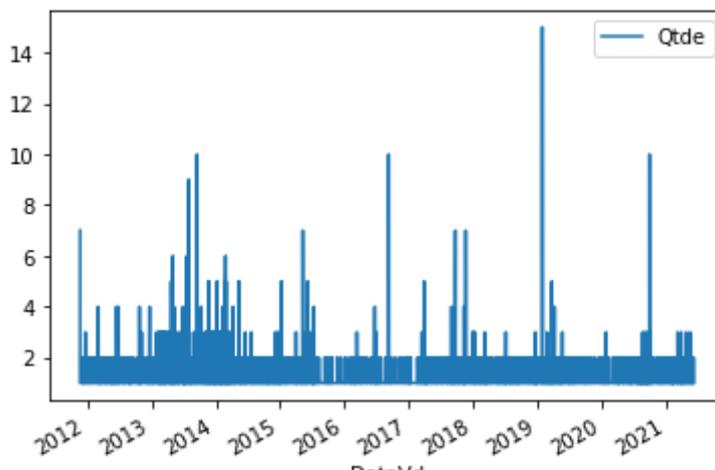
3.3. *Hardware*

Para o desenvolvimento dos experimentos, foi utilizado notebook Dell XPS com Processador i7 e 4 GB de memória RAM (*Random Access Memory*).

3.4. Caracterização e análise dos dados

A Figura 7 mostra o gráfico de vendas do produto pela empresa diariamente.

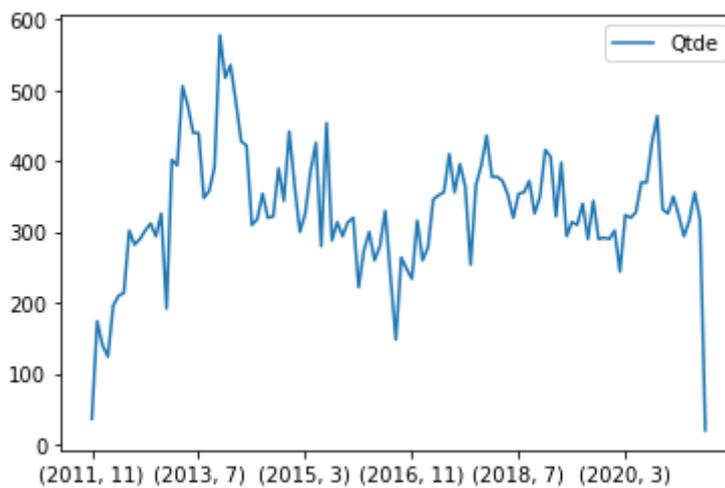
Figura 7 Vendas diariamente.



Fonte Próprio autor

De acordo com a Figura 7, é possível observar que as vendas diárias são inconsistentes. A Figura 8 mostra as vendas desse produto mensalmente.

Figura 8 Vendas mensalmente.



Fonte Próprio autor

De acordo com a Figura 8, a visualização em mês fica muito mais clara. Por isso foi escolhida a utilização dos dados agrupados mensalmente.

Uma predição diária é mais complexa, pois teria que fazer uma análise dia a dia do que vai vender. A realidade em que a empresa está inserida é que tenha uma predição mensal, para que possa já fazer o levantamento mensal do que será necessário para o próximo mês.

3.5. Python

O presente trabalho foi implementado utilizando a linguagem *Python* e suas bibliotecas para análise de dados e criação e rede neural.

Python é uma das linguagens mais comumente usadas na ciência de dados. Foi implementada na Holanda em 1989, e uma de suas principais características é que não é complicado de aprender.

Python é uma linguagem de código livre, multiplataforma e fácil de entender, pois utiliza poucos caracteres especiais, e possui uma identificação para marcação dos blocos, de fácil leitura e manutenção do código. Outro recurso da linguagem *Python* é que ela possui uma biblioteca padrão muito completa, incluindo métodos e funções de acesso a bancos de dados a interfaces gráficas de usuário.

Por esses e outros recursos, o *Python* acabou reunindo um grande grupo de especialistas em torno dele. Além de ajudar a resolver problemas levantados por membros da comunidade e responder a perguntas relacionadas à ciência e análise de dados, a comunidade também desenvolve, discute e compartilha questões relacionadas ao *Python*.

3.5.1. Bibliotecas

No algoritmo foi utilizado algumas bibliotecas, para ajudar na análise dos dados e na implementação da RNR, como a biblioteca *pandas* e *tensorflow*.

Pandas é uma biblioteca *Python* que fornece estruturas de dados rápidas, flexíveis e expressivas, projetadas para tornar o processamento de dados "relacionais" ou "rotulados" simples e intuitivo. Ele tem como objetivo ser o bloco de construção básico superior para a análise de dados prática e do

mundo real em Python. Uma das principais utilização dessa biblioteca no trabalho foi a armazenagem e análise dos dados.

TensorFlow é uma biblioteca de aprendizado de máquina de código aberto adequada para várias tarefas. É um sistema para criar e treinar redes neurais para detectar e decifrar padrões e correlações, semelhantes (mas não iguais) à maneira como os humanos aprendem e raciocinam. Ele é usado na pesquisa e produção do Google e gradualmente substituiu seu predecessor de código proprietário DistBelief.

4. Desenvolvimento

A rede neural recorrente que será usado será a **muita para muita**, com várias entradas e várias saídas. Primeiro, o algoritmo lê todos os dados de vendas citados no Apêndice A. O arquivo deve estar no formato .CSV e conter Três colunas: "mês", "ano" e "qtde". Cada coluna contém seus próprios dados, referentes a ao mês de vendas, ano de vendas e todas as vendas do mês atual. Então, o algoritmo irá pré-processar os dados e colocá-los em um *DataFrame* da biblioteca Pandas do *Python*.

A etapa de pré-processamento é responsável por criar vários vetores que serão usados como entrada para alimentar a rede neural recorrente. Finalmente, o algoritmo irá utilizar aa funções da biblioteca TensorFlow para realizar a previsão e treinar cada rede neural recorrente com 1 neurônio na camada de entrada. A camada de recorrência tem 64 neurônios e a camada de saída tem 1 neurônio.

4.1. Processamento do Algoritmo

O primeiro passo é organizar os dados importados na ordem crescente de mês e ano. Após, irá ser criado uma tabela cuja primeira coluna representa os dados reais das vendas. A cada nova coluna, vai possuir os dados de venda defasado em 1 período. Assim é criado uma tabela de sequencias para o treinamento da rede neural, sendo que a quantidade de linhas será a quantidade de registros. Foi retirado o primeiro e último registro por se tratar de *outliers*, a analise desses *outliers*, foi simples pois se tratava de meses que não teve vendas o mês todo.

A Tabela 1 mostra os valores criados na tabela de sequência para o treinamento.

Tabela 1 - Sequências para o treinamento.

Qtde	Qtde1	Qtde2	Qtde3	Qtde4	Qtde5	Qtde6	Qtde7	Qtde8	Qtde9	...	Qtde11	Qtde12	Qtde13	Qtde14	Qtde15	Qtde16	Qtde17	Qtde18	Qtde19	Qtde20
174	140.0	124.0	196.0	210.0	214.0	302.0	282.0	290.0	302.0	...	294.0	326.0	192.0	402.0	394.0	506.0	478.0	440.0	440.0	348.0
140	124.0	196.0	210.0	214.0	302.0	282.0	290.0	302.0	312.0	...	326.0	192.0	402.0	394.0	506.0	478.0	440.0	440.0	348.0	358.0
124	196.0	210.0	214.0	302.0	282.0	290.0	302.0	312.0	294.0	...	192.0	402.0	394.0	506.0	478.0	440.0	440.0	348.0	358.0	392.0
196	210.0	214.0	302.0	282.0	290.0	302.0	312.0	294.0	326.0	...	402.0	394.0	506.0	478.0	440.0	440.0	348.0	358.0	392.0	578.0
210	214.0	302.0	282.0	290.0	302.0	312.0	294.0	326.0	192.0	...	394.0	506.0	478.0	440.0	440.0	348.0	358.0	392.0	578.0	518.0
...
314	310.0	340.0	290.0	344.0	290.0	292.0	290.0	302.0	244.0	...	320.0	328.0	370.0	370.0	426.0	464.0	332.0	326.0	350.0	324.0
310	340.0	290.0	344.0	290.0	292.0	290.0	302.0	244.0	324.0	...	328.0	370.0	370.0	426.0	464.0	332.0	326.0	350.0	324.0	294.0
340	290.0	344.0	290.0	292.0	290.0	302.0	244.0	324.0	320.0	...	370.0	370.0	426.0	464.0	332.0	326.0	350.0	324.0	294.0	316.0
290	344.0	290.0	292.0	290.0	302.0	244.0	324.0	320.0	328.0	...	370.0	426.0	464.0	332.0	326.0	350.0	324.0	294.0	316.0	356.0
344	290.0	292.0	290.0	302.0	244.0	324.0	320.0	328.0	370.0	...	426.0	464.0	332.0	326.0	350.0	324.0	294.0	316.0	356.0	318.0

Fonte: Próprio autor

O próximo passo é separar os dados para o treino e dados para teste. A cada linha serão extraídas variáveis independentes. Essas variáveis vão da coluna 0 até a coluna N-1, sendo que N representa o número de períodos selecionados. Nesse processo serão extraídas as variáveis dependentes que vai de 1 até N, e assim vai definir a amostra que vai ser utilizada para teste e para treino.

Após o algoritmo capturar os valores para treino e teste da RNR, o algoritmo usa a biblioteca *Tensorflow* responsável pela criação e treinamento da RNR. É informado os valores dela, sendo vai ter 1 *input*, 64 neurônios na camada oculta de recorrência e 1 *output*.

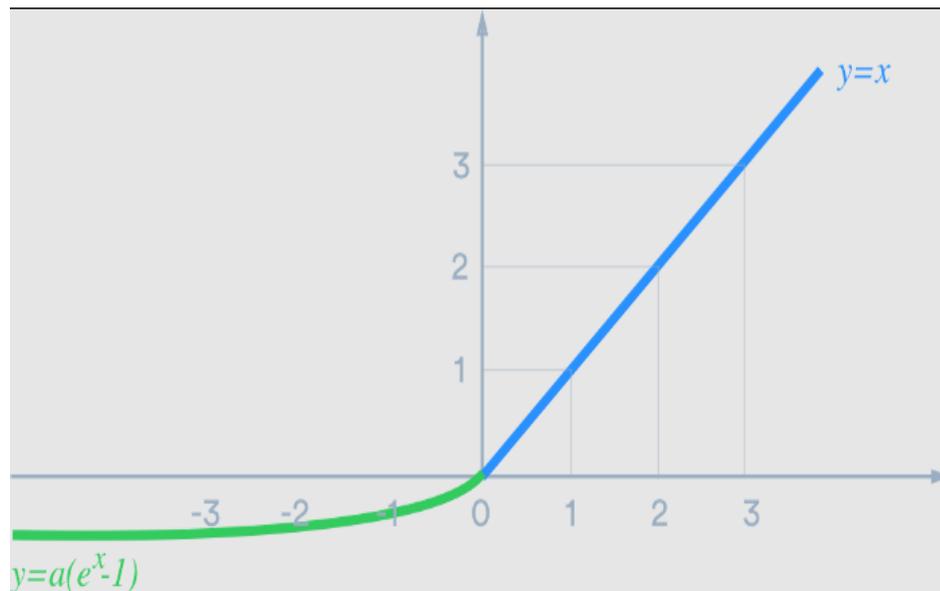
Depois de definir alguns hiper parâmetros do modelo e abrir o gráfico do TensorFlow, começamos com `tf.placeholder (...)`, que será onde forneceremos os dados para o gráfico. A seguir, usamos `BasicRNNCell (...)` para definir o estado do loop. Você pode pensar neste objeto como uma "fábrica" para a criação de cópias de unidades recorrentes, ou seja, desdobrando-se na dimensão do tempo. Em seguida, passamos essa "fábrica" junto com o marcador de posição de entrada para a função `tf.nn.dynamic_rnn (...)`. Em seguida, ele cria vários estados ocultos, um para cada período, e os conecta usando a multiplicação de matrizes. `tf.nn.dynamic_rnn (...)` retorna uma saída para cada período de tempo e o último estado oculto da rede. Nesse caso, saídas `[:, -1, :]` = `last_state`, ou seja, o último estado é igual à última saída.

Para a ativação da camada oculta de recorrência é utilizada a função de ativação *Exponential Linear Unit (ELU)* proposta em 2015 por Djork-Arné Clevert

et al., representada pela Equação (2) e a Figura 9 mostra a sua representação gráfica.

$$R(z) = f(x) = \begin{cases} z, z > 0 \\ \alpha_1(e^z - 1), z \leq 0 \end{cases} \quad (2)$$

Figura 9 função de ativação ELU.



Fonte (Liu, 2017)

De acordo com a Figura 9, para valores menores do que 0, o valor da função é definido por uma exponencial. Para valores acima de 0, a função se torna linear.

Foi escolhido como erro médio absoluto (EMA) esta medida se afasta de zero à medida que as discrepâncias entre a previsão e a observação aumentam do custo. Isso ocorre porque as séries temporais de demanda costumam ter picos e vales muito acentuados que entram em conflito com a tendência geral. Como o modelo não é sensível a esses pontos extremos, escolhemos uma métrica robusta para outliers.

No algoritmo foram realizadas 10000 iterações. Tais iterações foram otimizadas utilizando o AdamOptimizer que é uma extensão da descida do gradiente estocástico e pode ser usada no lugar da descida do gradiente estocástico clássico para atualizar os pesos da rede com mais eficiência, utilizando lotes de 64 minilotes cada. O erro médio absoluto referente ao treinamento foi calculado a cada 2000 iterações.

Ao final foi calculado o coeficiente de determinação (R^2) através da biblioteca *sklearn.metrics* que chama a função `r2_score(y_true, y_pred)`, *y_true* a matriz com valores reais, e *y_pred* a matriz com valores previstos, que descreve a fração da variância total nos dados observados que pode ser explicada pelo modelo. Seu valor fica entre 0 e 1, e quanto maior o valor, melhor a concordância entre modelo e observação.

5. RESULTADOS

Após o desenvolvimento do algoritmo que implementa uma RNR, foram realizados os testes utilizando os dados de vendas, 12 passos de tempo para treinamento, 117 registros para treinamento da RNR e com uma taxa de aprendizado de 0.001. A Tabela 2 apresenta os resultados obtidos de Erro Médio Absoluto (EMA) para conjuntos de iterações.

Tabela 2: Erro médio absoluto a cada 2000 iterações e erro médio absoluto total do teste

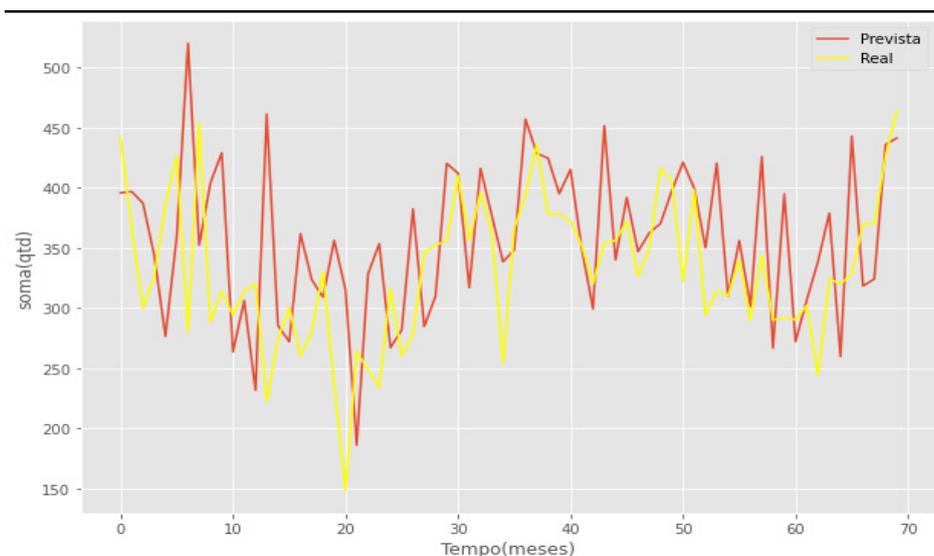
Iterações	Erro médio absoluto %
0	525,3625
2.000	22,162022
4.000	16,62214
6.000	14,3411
8.000	13,4641695
10.000	12,674756

Fonte: Próprio autor

De acordo com a Tabela 2, é possível notar que no final do treinamento (após 10.000 iterações) o EMA é de 12,674756%. O coeficiente de determinação (R^2) calculado foi de -0.4750891892915472.

A Figura 10, mostra o resultado do treinamento da RNR em um gráfico de linhas onde a linha vermelha é o resultado previsto de vendas e a linha amarela é o valor real de venda em cada mês.

Figura 10 Resultado experimental do treinamento da rede neural recorrente utilizando dados de vendas.



Fonte Próprio Autor

De acordo com a Figura 10, no início da previsão os valores são diferentes do real. Porém, ao passar dos períodos, os valores previstos ficam parecidos, assemelhando a um cenário real. Desta forma, os valores não gerariam uma perda ao empresário.

Ainda de acordo com a Figura 10, é possível verificar que períodos de altas em vendas repentinas, ou em quedas altas, o modelo proposto retorna um erro muito alto na previsão. No cenário atual, devido a pandemia do Corona Vírus, alguns produtos poderiam ter sua previsão afetada, fazendo com que o empresário ficasse com um estoque muito alto ou até com falta de estoque de determinados produtos.

6. CONCLUSÃO

É notável que atualmente existem grandes quantidades de dados. O estudo adequado sobre esses dados, traz benefícios a empresas. Um dos problemas de empresários no varejo, em geral, é o controle de estoque. Fazer uma previsão assertiva de estoque é possível otimizar capital e evitar perdas. Este trabalho teve como objetivo a utilização de mineração de dados no desenvolvimento de um sistema de previsão de demanda de estoque.

Diante deste contexto levantado no trabalho é possível verificar como é importante o trabalho de análise e previsão de dados em um varejo brasileiro, já que é de extrema importância que o gestor tenha em mãos esses dados, para futuras tomadas de decisões.

Após estudos e análises foi verificado que os algoritmos de RNR são bem robustos e eficazes para previsões de demandas além de serem rápidos. Além disso, além de ser bastante mutável para vários tipos de produtos e áreas.

O modelo proposto de RNR foi validado através de diversos experimentos empíricos, observando os dados reais e os dados previstos no modelo. O modelo obteve um resultado satisfatório, com valor de EMA aceitável de 12,674756%, indicando o tamanho do erro. Porém o coeficiente de determinação calculado resultou em um valor negativo, sendo que o valor perfeito seria próximo a 1.

Como trabalho futuro sugere-se a melhoria no modelo como, por exemplo, automatizar os *inputs*. Além disso, adicionar mais dados na previsão e verificar produtos com vendas sazonais e os motivos que esses produtos tenham maior vendas apenas em alguns períodos, para assim conseguir ter uma previsão mais assertiva.

REFERÊNCIAS

GRÜBLER, MURILLO. **Entendendo o funcionamento de uma Rede Neural Artificial**. Medium. Disponível em: <<https://medium.com/brasil-ai/entendendo-o-funcionamento-de-uma-rede-neural-artificial-4463fcf44dd0>>. Acesso em: 22 May 2021.

LANHELLAS, RONALDO. **Redes Neurais Artificiais: Algoritmo Backpropagation**. DevMedia. Disponível em: <<https://www.devmedia.com.br/redes-neurais-artificiais-algoritmo-backpropagation/28559>>. Acesso em: 27 May 2021.

TRASK, I AM **Anyone Can Learn To Code an LSTM-RNN in Python (Part 1: RNN)**. iamtrask.github.io. Disponível em: <<https://iamtrask.github.io/2015/11/15/anyone-can-code-lstm/>>. Acesso em: 30 May 2021.

The Unreasonable Effectiveness of Recurrent Neural Networks. Github.io. Disponível em: <<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>>. Acesso em: 30 May 2021.

LIU, DANQING. **A Practical Guide to ReLU - Danqing Liu - Medium**. Medium. Disponível em: <<https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>>. Acesso em: 2 Jun. 2021.

Facure, Matheus. RNRs para Séries de Tempo. Github.io. Disponível em: <<https://matheusfacure.github.io/2017/09/27/rnn-tf-ts/>>. Acesso em: 2 Jun. 2021. Andrej Karpathy

Karpathy, Andrej. **The Unreasonable Effectiveness of Recurrent Neural Networks**. Github.io. Disponível em: <<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>>. Acesso em: 2 Jun. 2021.

VITÓRIA, ARTHUR. **Classificação De Bactérias E Determinação Da Coloração De Gram Utilizando Imagens Hiperespectrais E Aprendizado De**

Máquina. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/handle/123456789/604>>. Acesso em 2 Jun 2021.

PEINALDO, J. **Administração da Produção**. [S.l.]: [s.n.], 2007.

SEBRAE, S. B. D. A. À. E. E. P. E.-. **Causa Mortis**. São Paulo: Atlas, 2014.

RITZMAN, L. P.; KRAJEWSKI, L. J.; KLASSEN, R. **Foundations of operations management**. Toronto: Pearson Prentice Hall, 2004.

FEIJÓ, Bruno Vieira. **A Revolução dos Dados**. *Revista Exame PME – Pequenas e Médias Empresas*, São Paulo, p. 30-43, set. 2013. Acesso em 23 mai. 2020>.

FAYYAD, U.M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **From Data Mining to Knowledge Discovery**. *AI Magazine*, 1996, p. 37-54. .

FAYYAD, U. M., Piatetsky Shapiro, G., Smyth, P. & Uthurusamy, R. “**Advances in Knowledge Discovery and Data Mining**” 1996, AAAIPress, The Mit Press.

PASSARI, Lima ; FABRIZIO, Antônio. **Previsão de vendas no varejo por meio de redes neurais**. *Revista Adm.* v41, p257-272, jul-ago-set-2006.

CATARINO, MARIANO, **Python: melhor linguagem para Machine Learning. Entenda!** Blog Impacta - Dicas e Notícias sobre Carreira. Disponível em: <<https://www.impacta.com.br/blog/python-melhor-linguagem-para-machine-learning-entenda/>>. Acesso em: 15 Jun. 2021.

PYKES, Kurtis. **Adam Optimization Algorithm - Towards Data Science**. Medium. Disponível em: <<https://towardsdatascience.com/adam-optimization-algorithm-1cdc9b12724a>>. Acesso em: 15 Jun. 2021.

CONTRIBUIDORES DOS PROJETOS DA WIKIMEDIA. **biblioteca de software de aprendizagem de máquina**. Wikipedia.org. Disponível em: <<https://pt.wikipedia.org/wiki/TensorFlow>>. Acesso em: 15 Jun. 2021.

**APÊNDICE A – DADOS DE VENDAS MENSIS DE
MEDICAMENTO ENTRE NOVEMBRO DE 2011 E JUNHO DE
2021**

Mes	Ano	Quantidade vendida mensalmente
11	2011	36
12	2011	174
1	2012	140
2	2012	124
3	2012	196
4	2012	210
5	2012	214
6	2012	302
7	2012	282
8	2012	290
9	2012	302
10	2012	312
11	2012	294
12	2012	326
1	2013	192
2	2013	402
3	2013	394
4	2013	506
5	2013	478
6	2013	440
7	2013	440
8	2013	348
9	2013	358
10	2013	392
11	2013	578
12	2013	518
1	2014	536
2	2014	484
3	2014	428
4	2014	422
5	2014	310
6	2014	318
7	2014	354
8	2014	320
9	2014	322
10	2014	390

11	2014	344
12	2014	442
1	2015	366
2	2015	300
3	2015	326
4	2015	386
5	2015	426
6	2015	280
7	2015	454
8	2015	288
9	2015	314
10	2015	294
11	2015	314
12	2015	320
1	2016	222
2	2016	274
3	2016	300
4	2016	260
5	2016	280
6	2016	330
7	2016	236
8	2016	148
9	2016	264
10	2016	248
11	2016	234
12	2016	316
1	2017	260
2	2017	278
3	2017	346
4	2017	352
5	2017	356
6	2017	410
7	2017	356
8	2017	396
9	2017	364
10	2017	254
11	2017	366
12	2017	394
1	2018	436
2	2018	378
3	2018	378
4	2018	372
5	2018	352
6	2018	320
7	2018	354

8	2018	356
9	2018	372
10	2018	326
11	2018	350
12	2018	416
1	2019	406
2	2019	322
3	2019	398
4	2019	294
5	2019	314
6	2019	310
7	2019	340
8	2019	290
9	2019	344
10	2019	290
11	2019	292
12	2019	290
1	2020	302
2	2020	244
3	2020	324
4	2020	320
5	2020	328
6	2020	370
7	2020	370
8	2020	426
9	2020	464
10	2020	332
11	2020	326
12	2020	350
1	2021	324
2	2021	294
3	2021	316
4	2021	356
5	2021	318
6	2021	20



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
PRÓ-REITORIA DE GRADUAÇÃO

Av. Universitária, 1069 • Setor Universitário
Caixa Postal 86 • CEP 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946.1021 | Fax: (62) 3946.1397
www.pucgoias.edu.br | prograd@pucgoias.edu.br

ANEXO I APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O estudante Lucas de Souza Ferreira do Curso de Ciências da Computação, matrícula 2015.1.0028.0035-7, telefone (62) 98501-0677, e-mail lucassf54@gmail.com, na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Controle de Estoque Utilizando Redes Neurais, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros), específicos da área para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 08 de junho de 2021

Assinatura do autor:

Nome completo do autor:

Lucas de Souza Ferreira

Assinatura do professor – orientador:

Nome completo do professor – orientador:

Gustavo Siqueira Vinhal