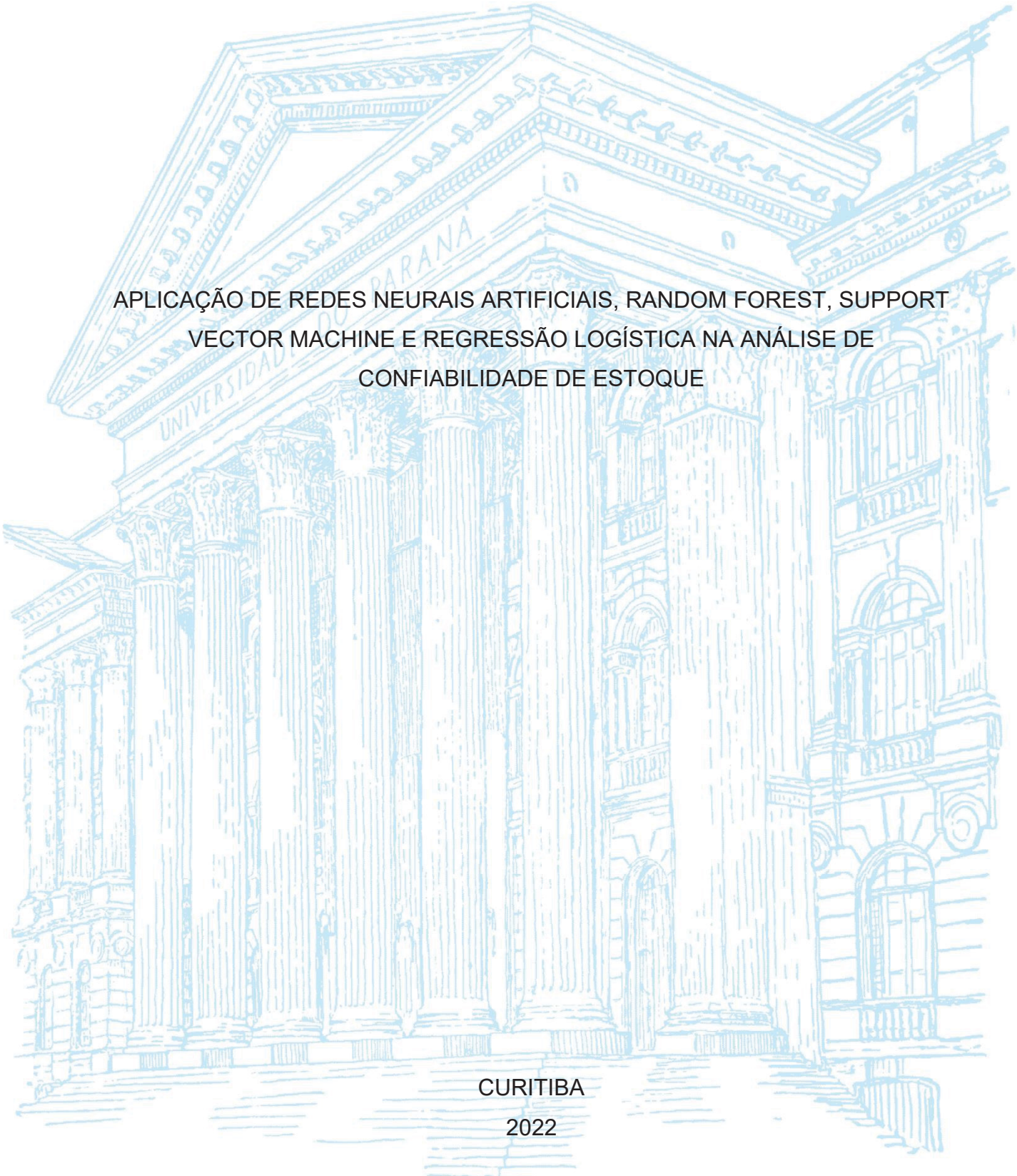
UNIVERSIDADE FEDERAL DO PARANÁ

RAUL VALDERA JUNIOR



APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS, RANDOM FOREST, SUPPORT VECTOR MACHINE E REGRESSÃO LOGÍSTICA NA ANÁLISE DE CONFIABILIDADE DE ESTOQUE

CURITIBA

2022

RAUL VALDERA JUNIOR

APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS, RANDOM FOREST, SUPPORT VECTOR MACHINE E REGRESSÃO LOGÍSTICA NA ANÁLISE DE CONFIABILIDADE DE ESTOQUE

Trabalho de conclusão de curso apresentado ao curso de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial.

Orientador: Prof. Dr. Jaime Wojciechowski

CURITIBA

2022

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **RAUL VALDERA JUNIOR** intitulada: **Aplicação De Redes Neurais Artificiais, Random Forest, Support Vector Machine E Regressão Logística Na Análise De Confiabilidade De Estoque**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 08 de Novembro de 2022.

JAIME WOJCIECHOWSKI
Presidente da Banca Examinadora

RAZER ANTHOM NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Aplicação de Redes Neurais Artificiais, *Random Forest, Support Vector Machine* e Regressão Logística na análise de confiabilidade de estoque

Raul Valdera Junior
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
valderajr@gmail.com

Jaime Wojciechowski
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
jaimewo@ufpr.br

***Resumo*— Com a constante expansão do comércio eletrônico, surgiram também novas soluções e oportunidades de melhoria para problemas não tão frequentes, ou até mesmo inexistentes, nos comércios de lojas exclusivamente físicas. *Marketplaces*, por exemplo, acabam sendo uma ponte digital entre o vendedor e o comprador. Entretanto, para esse sistema de parceria digital funcionar, todas as informações fornecidas pelo vendedor precisam ser precisas e atualizadas, principalmente com relação à quantidade de estoque disponível. Quando a informação não condiz com a realidade e um pedido é gerado, acaba resultando em cancelamentos, multas, penalizações na reputação da loja e frustração dos compradores. Neste sentido, este trabalho traz uma análise comparativa do poder preditivo de classificação da veracidade das informações de estoque fornecidas, aplicando-se, para isso, os modelos de RNN, RF, SVM e regressão logística. Verifica-se, então, que para essa abordagem, o modelo RF apresenta os melhores resultados com relação à acurácia e especificidade.**

Palavras-chave— classificação, análise, estoque, RNN, RF, SVM, logística, regressão

***Abstract*— With the constant expansion of e-commerce, new solutions and improvement opportunities have also emerged for problems that are not so often or even existent physical stores. Marketplaces, for example, turned into a digital bridge between the seller and the buyer. However, for this digital partnership system to work, all information provided by the seller needs to be accurate and up-to-date, especially regarding the amount of available stock. When the information does not match reality and an order is created, it results in cancellations, fines, penalties on the store's reputation and frustration for buyers This work presents a comparative analysis of the predictive power of classifying the veracity of the stock information provided, applying, for this, the RNN, RF, SVM and logistic regression models. It is verified, then, that for this approach, the RF model presents the best results regarding accuracy and specificity.**

Keywords— classification, analysis, stock, RNN, RF, SVM, logistic, regression

I. DESENVOLVIMENTO

Nos últimos anos, com o crescente desenvolvimento do comércio eletrônico (*e-commerce*), surgiram e continuam surgindo novos meios de se vender *online* ou de facilitar a venda nos modelos de negócio já existentes.

Entre esses modelos de negócio, por exemplo, estão os *marketplaces*. Nessas estruturas, uma determinada empresa ou organização desenvolve uma plataforma que serve de vitrine para os vendedores anunciarem seus produtos, receber ordens de compra e controlar o seu estoque [9].

O estoque, por sua vez, é uma informação que necessita ser precisa, pois quando consta que há estoque na plataforma e, ao ser gerada uma ordem de compra, verifica-se que o estoque acabou, gera prejuízos tanto financeiros quanto de imagem da marca ou loja.

Visando buscar soluções para identificar esse possível problema de confiabilidade de estoque, classificando-os em confiáveis ou não, modelos de redes neurais artificiais (RNA), *random forest* (RF), *support vector machine* (SVM) e regressão logística (RL) foram utilizados. As técnicas de modelagem em questão foram aplicadas em uma base de empresa privada com dados de ordens de compra, utilizando-se o R como linguagem de programação.

A. Descrição dos dados

A base de dados utilizada foi cedida pela empresa Olist Serviços Digitais SA. Originalmente ela continha 548.941 registros e 34 variáveis, conforme descritos na Tabela I. Entretanto, por haver registros apresentando a maior parte dos seus atributos como *missing*, variáveis de identificação das transações e campos que não fariam sentido na análise por serem informações referentes aos compradores ao invés dos lojistas e rastreamento dos pedidos, um processo de tratamento dos dados foi aplicado.

TABELA I. TABELA DE ATRIBUTOS DA BASE DE DADOS ORIGINAL

Atributo	Descrição	Tipo
order_id	identificador da ordem de compra	Int
purchase_timestamp	data da ordem de compra	Date
origin	UF de origem do comprador	Text
brand	marca do produto	Text
creation_date_product	data de criação do produto na base	Date
creation_date_seller	data de criação do usuário do lojista na base	Date
gmv	valor da ordem de compra	Dec
orders	número de produtos comprados	Int
status	status da ordem de compra	Text
gmv_u30d	receita gerada pelo produto nos últimos 30 dias antes da ordem de compra	Dec
orders_u30d	vendas do produto nos últimos 30 dias antes da ordem de compra	Int

Atributo	Descrição	Tipo
gmv_u60d	receita gerada pelo produto nos últimos 60 dias antes da ordem de compra	Dec
orders_u60d	vendas do produto nos últimos 60 dias antes da ordem de compra	Int
nps_score	nota da última avaliação dada pelo lojista	Dec
nps_date	data da última avaliação	Date
seller_type	lojista com estoque no fulfillment da empresa (FBO) ou não (3P)	Text
category_catalog	categoria do produto	Text
type_catalog	subcategoria do produto	Text
banda	grau de representatividade interna do produto	Text
tag_area	área que atende o lojista	Text
plan_type	plano contratado pelo lojista	Text
first_order	data da primeira venda do produto	Date
last_order	data da última venda do produto	Date
stock_date	última atualização do estoque	Date
stock	quantidade no estoque	Int
last_order_issue	estoque incorreto na venda anterior	Text
active_campaign	campanha ativa no momento da venda	Text
gtin_score	grau de representatividade do produto no mercado	Dec
is_manufacturer	lojista é fabricante ou não	Text
in_logistica_centralization	lojista utiliza coleta da empresa de e-commerce	Text
seller_index	índice de qualidade do lojista	Dec
has_integration	lojista utiliza integrador de estoque ou não	Text
address_code	2 primeiros dígitos (ou 1 quando o primeiro for 0) do cep do lojista.	Int
stock_issue	variável resposta já categorizada indicando estoque incorreto ou não	Text

Com a limpeza da base e criação de variáveis adicionais baseadas, em sua maioria, nas diferenças entre os atributos de data, resultaram 496.460 registros e 23 atributos, já incluindo a variável resposta já marcada desde a base original. As variáveis retiradas da análise, bem como o respectivo motivo baseado sempre na lógica de negócio da operação, encontram-se na Tabela II. As descrições dos atributos desta base resultante estão descritas na Tabela III.

No que diz respeito à variável resposta, esta classifica a ocorrência (*Yes*) ou não (*No*) de problemas relacionados ao estoque na ordem de compra. O objetivo do estudo em questão foi então o de utilizar as variáveis já filtradas e ajustadas da Tabela III, baseadas no histórico de pedidos já realizados, para identificar características, sejam do lojista, do produto ou do estoque propriamente dito, que indiquem que o estoque informado pelo lojista não condiga com a realidade. Nessa situação, caso houvesse a realização de um pedido, ele iria apresentar algum problema. Sendo assim, ações preventivas poderiam ser tomadas antes da realização desses pedidos a fim de reduzir os custos com multas, bem como manter a reputação da loja alta dentro do *marketplace*.

Considerando que os recursos computacionais para se trabalhar com uma base não eram suficientes, optou-se por utilizar uma amostra aleatória, sem reposição, de 40 mil registros aplicando a função *sample,* nativa do R. Destes registros, aproximadamente 1,41% apresentaram problemas com a veracidade do estoque fornecido pelo lojista na plataforma da empresa. Em outras palavras, a cada 200 casos, aproximadamente 3 apresentam algum problema. Este fato caracteriza a existência de um desbalanceamento na base com relação às proporções das classes da variável resposta, podendo comprometer a análise. Entretanto, como o intuito é a comparação das técnicas de treinamento, não foram realizados tratamentos com relação ao desbalanceamento.

Por questões de sigilo e confidencialidade, as informações sensíveis de identificação das ordens de compra e/ou produtos em questão foram retiradas e valores arbitrários foram atribuídos. Tal medida, entretanto, não invalidou as comparações realizadas, pois os atributos impactados não foram utilizados como variáveis explicativas nos modelos.

TABELA II. VARIÁVEIS RETIRADAS DA BASE ORIGINAL

Atributo Retirado	Motivo
order_id	Por ser um registro único, não pode ser utilizado como variável pois o modelo poderia "decorar" os dados
brand	Variabilidade muito grande de classes
type_catalog	
category_catalog	
tag_area	
status	Essas variáveis são criadas depois que a ordem é gerada e o objetivo do estudo é prever que uma ordem de compra terá problemas com estoque antes que aconteça e impedir que ocorra de fato
gmv	
orders	
origin	
nps_date	Variável com mais de 50% dos valores *missing*
seller_type	Utilizada como auxiliar para a criação da variável binária is_fulfillment
creation_date_product	Utilizada como auxiliar para a criação da variável product_age
creation_date_seller	Utilizada como auxiliar para a criação da variável seller_age
stock_date	Utilizada como auxiliar para a criação da variável days_wo_updt_stock
last_order	Utilizada como auxiliar para a criação da variável days_wo_purchase
first_order	Utilizada como auxiliar para a criação da variável days_frst_purchase
purchase_timestamp	Utilizada como auxiliar para a criação das variáveis product_age, seller_age, days_wo_updt_stock, days_wo_purchase e days_frst_purchase

TABELA III. TABELA DE ATRIBUTOS DA BASE DE TREINO

Atributo	Descrição	Tipo
gmv_u30d	receita gerada pelo produto nos últimos 30 dias antes da ordem de compra	Dec
orders_u30d	vendas do produto nos últimos 30 dias antes da ordem de compra	Int
gmv_u60d	receita gerada pelo produto nos últimos 60 dias antes da ordem de compra	Dec
orders_u60d	vendas do produto nos últimos 60 dias antes da ordem de compra	Int

Atributo	Descrição	Tipo
nps_score	nota da última avaliação dada pelo lojista	Dec
is_fulfillment	lojista com estoque no fulfillment da empresa ou não	Int
banda	grau de representatividade interna do produto	Int
plan_type	plano contratado pelo lojista	Text
stock	quantidade no estoque	Int
last_order_issue	estoque incorreto na venda anterior	Int
active_campaign	campanha ativa no momento da venda	Int
gtin_score	grau de representatividade do produto no mercado	Dec
is_manufacturer	lojista é fabricante ou não	Int
in_logistica_centralization	lojista utiliza coleta da empresa de e-commerce	Int
seller_index	índice de qualidade do lojista	Dec
has_integration	lojista utiliza integrador de estoque ou não	Int
address_code	2 primeiros dígitos (ou 1 quando o primeiro for 0) do cep do lojista.	Int
product_age	dias desde o cadastro do produto na base	Int
seller_age	dias desde o cadastro do usuário do lojista na base	Int
days_wo_updt_stock	dias sem atualização de estoque	Int
days_wo_purchase	dias sem venda do produto	Int
days_frst_purchase	dias desde a primeira venda do produto	Int
stock_issue	variável resposta já categorizada indicando estoque incorreto ou não	Text

B. *Métodos*

Para que seja possível a compreensão dos modelos de classificação empregados, primeiramente deve-se entender o funcionamento do fluxo simplificado de uma compra em *marketplaces*, conforme apresentado na Fig. 1.

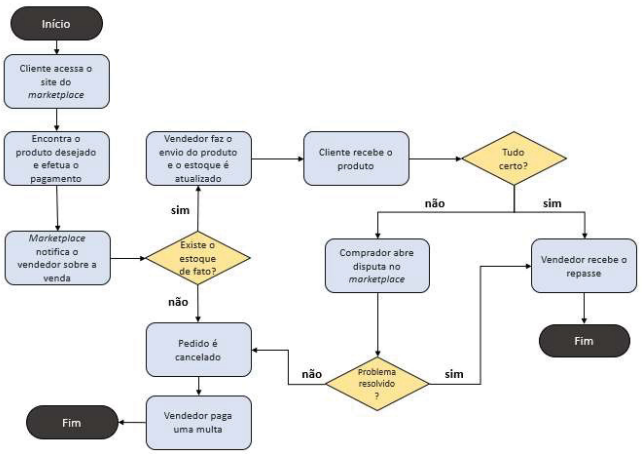


Fig. 1. Fluxo de Venda em um Marketplace.

Esse fluxo tem início com o possível comprador acessando algum *site* de *marketplace online*, se interessando pelo produto e efetuando o pagamento. Nesse momento, é gerada uma ordem de venda e o lojista é acionado para que prepare o envio do produto. Entretanto, quando ele informa que não tem estoque, seja pelo número desatualizado ou mesmo por preencher de forma equivocada, cancela-se o pedido e uma multa é aplicada para a plataforma que informou o estoque. Nos casos sem nenhum problema com estoque, segue o fluxo normalmente, ou seja, se o cliente receber o produto, for conforme o anunciado e em perfeitas condições, é feito o repasse ao lojista. Caso contrário, tenta-se resolver qualquer eventual problema e, em último caso, o pedido também é cancelado.

Nota-se, então, que ter estoque disponível na plataforma não implica necessariamente em ter o estoque de fato com o vendedor, visto que a veracidade da informação depende exclusivamente dele.

Levando-se em consideração que atualmente existem diversas técnicas de inteligência artificial focadas na classificação de indivíduos ou situações, além de haver uma marcação na base de dados de problema relacionado ao estoque com a ordem de compra, optou-se pela utilização de modelos de classificação supervisionados.

Entre os classificadores mais tradicionais, é possível destacar o *Random Forest*, que são combinações dos preditores de árvores de decisão [1]. Além do RF, é possível a utilização de SVM, classificador binário que recebe os vetores de entrada e os mapeia não linearmente em um espaço de alta dimensão [4], redes neurais artificiais, que simulam o modelo de aprendizado dos neurônios humanos. Outra abordagem utilizada também foi a aplicação do modelo estatístico de regressão logística. Segundo [11], a análise de regressão logística múltipla, muito utilizada entre profissionais da área de análise de crédito, por exemplo, não apresenta grandes problemas de pressuposições, como a necessidade de uma distribuição Normal das variáveis explicativas. Ainda, de acordo com [11], essa abordagem apresenta facilidade computacional, visto que já está presente nos pacotes estatísticos mais utilizados.

1) *Pré-processamento dos dados:* Antes do treinamento, contudo, além dos tratamentos e criação de variáveis já mencionados na descrição da base de dados, fez-se também uma análise de correlação entre as variáveis preditivas, utilizando as funções *cor* e *corrplot* do pacote *corrplot*, a fim de evitar possíveis vieses do modelo por questões de multicolinearidade. Este problema, também conhecido como autocorrelação entre as variáveis independentes, pode ser verificado calculando-se o coeficiente de correlação de Person entre duas variáveis e valores maiores que 0,8 ou 0,9 podem indicar um problema sério de multicolinearidade [12]. Como é possível de se observar na Fig. 2, algumas variáveis apresentam forte autocorrelação entre si e, sendo assim, também foram retiradas da análise. Foram elas: *orders_u30d*, *orders_u60d*, *days_wo_updt_stock* e *days_frst_purchase*.

Além da retirada das variáveis com autocorrelação acentuada, aplicou-se uma normalização em todas as variáveis numéricas, fazendo-se uso da função *min-max*. Com essa normalização, o domínio das variáveis fica restrito entre 0 e 1. Essa normalização auxilia na conversão dos algoritmos empregados, pois como a variação é menor, o consumo de memória durante o processamento também é reduzido, permitindo a aplicação dos modelos em bases maiores, com milhares de registros, por exemplo [8].

2) *Preparação do treinamento:* No que diz respeito ao treinamento dos modelos mencionados, realizou-se um *holdout*, sendo a base de dados separada em 80% dos registros

para treinamento e 20% para teste, mantendo-se a proporção de ordens com e sem problema de estoque em ambas as partes. Para essa divisão, no pacote *caret* do R existe a função *createDataPartition*, com a qual é possível especificar quais variáveis devem manter-se balanceadas, baseando-se nos percentis. Nessa primeira etapa, um treinamento inicial de cada modelo é feito, onde o software R atribui os *grids* e testa automaticamente os hiperparâmetros para os modelos.

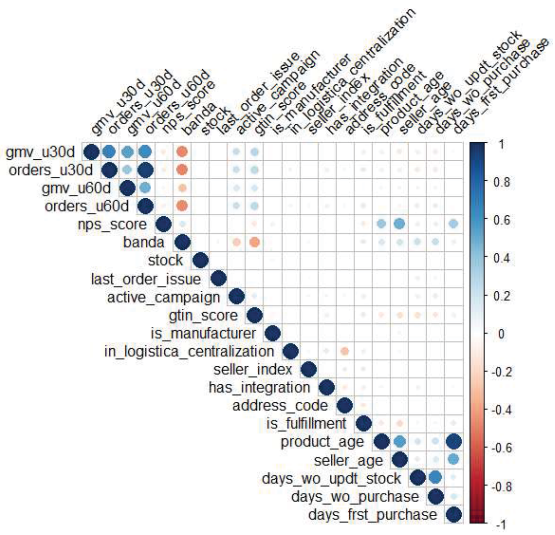


Fig. 2. Autocorrelação Entre Variáveis Preditoras.

Na fase de testes final, com o *tuning* de hiperparâmetros, fez-se uso do método *cross-validation*, com 10 partições (*folds*). Essa técnica consiste em uma abordagem alternativa para uma partição fixa, na qual várias partições são geradas, potencialmente permitindo que cada amostra seja usada várias vezes para vários propósitos, com o objetivo geral de melhorar a confiabilidade estatística dos resultados [10].

3) *Treinamento:* Para o treinamento dos modelos, os *grids* de hiperparâmetros variaram de acordo com cada abordagem, embasados nos valores resultantes dos treinamentos iniciais realizados com o *holdout*.

No modelo RNA, os hiperparâmetros em questão são a quantidade de neurônios na camada oculta da rede (*size*) e o decaimento dos pesos (*decay*). Para o hiperparâmetro *size*, o *grid* de valores variou de 1 a 10 com incrementos a cada 1. Para o *decay*, por sua vez, a variação foi de 0 a 0,5 com incrementos de 0,1.

No modelo SVM, os seus hiperparâmetros são o C e o σ (sigma). Para C, o grid de valores foi de 1 a 5, com incrementos a cada 1. Com relação ao σ, a variação foi de 0 a 0,3, com incrementos de 0,05.

No modelo RF, os hiperparâmetros são o número de árvores (*ntree*) e o número de características (*mtry*). Para *ntree*, a variação foi de 500 a 2.000, com incrementos a cada 500. Por fim, para *mtry* os valores variaram de 1 a 6, com incrementos a cada 1.

Para o treinamento do modelo RL utilizou-se o método conhecido como *stepwise,* que se trata de uma combinação dos métodos *forward* e *backward*, onde em cada etapa do processo de seleção, variáveis podem ser adicionadas ou subtraídas [7][13]. No modelo final, as variáveis foram significativas aos níveis de 0.1%, 1%, 5% e 10%. Ao contrário dos demais modelos, o resultado regressão logística é a probabilidade do de ocorrer problemas com relação ao estoque e não a classificação em si. Por esse motivo, aplicou-se um ponto de corte para definição das classes foi de 0,5, ou seja, valores de probabilidade abaixo de 50% são classificados como "Não", indicando que não ocorreriam problemas de estoque e, valores acima de 50%, indicando que ocorreriam problemas na ordem de compra.

Fez-se uso também, em virtude das limitações técnicas de processamento na máquina utilizada para o treinamento dos modelos, das funções *makeCluster* e *registerDoParallel*, dos pacotes *parallel* e *doParallel*, respectivamente. Tais funções auxiliaram na paralelização do processo entre todos os núcleos do processador.

Por fim, para definir a qualidade dos ajustes dos modelos, fez-se uso de uma métrica já tradicionalmente usada em modelos de classificação: a acurácia [2]. O cálculo da acurácia foi baseado nas matrizes de confusão, onde:

$$acc = (VN + VP)/(VN + VP + FN + FP) \quad (1)$$

sendo VN os verdadeiros negativos, VP os verdadeiros positivos, FN os falsos negativos e FP os falsos positivos.

Em adição à acurácia, principalmente pelo fato da variável resposta ser desbalanceada na proporção entre as duas classificações possíveis, optou-se por analisar também a sensibilidade e a especificidade, calculadas segundo as equações a seguir:

$$sens = VP/(VP + FN) \quad (2)$$

$$spec = VN/(VN + FP) \quad (3)$$

A sensibilidade caracteriza-se por ser a capacidade de discriminar, dentre suspeitos de um determinado fenômeno de estudo, aqueles que realmente apresentam o fenômeno de fato. A especificidade, por sua vez, é a capacidade de detectar os que realmente não tem relação com o fenômeno [3]. No estudo em questão, esse fenômeno é a ocorrência ou não de problemas relacionados ao estoque em uma ordem de compra.

Com essas medidas adicionais, além da assertividade do modelo de um modelo geral, obtido através da acurácia, pode-se analisar também o percentual de acerto em cada uma das classes, ou seja, o acerto ao se comparar predito e realizado dos verdadeiros positivos e verdadeiros negativos. Como a base é desbalanceada, analisar os casos em que houve a identificação correta das ordens com problema de estoque pode ser mais interessante, em termos práticos, do que a acurácia geral.

C. Tecnologias

A análise de comparação dos modelos preditivos foi desenvolvida em um computador pessoal Dell Latitue 3490, com processador i7 de 8ª geração e 1,8 GHz, com 16 *gigabytes* de memória RAM, sendo o sistema operacional Windows 10 versão 21H1. No desenvolvimento do *script*, foram utilizadas as seguintes aplicações e bibliotecas:

- Linguagem R, versão 4.2.1;
- Interface RStudio, versão 2022.07.1+554;
- Pacote *caret*, versão 6.0.93;
- Pacote *rlang*, versão 1.0.6;
- Pacote *dplyr*, versão 1.0.10;
- Pacote *corrplot*, versão 0.92;

• Pacote *parallel*, versão 4.2.1;
• Pacote *doParallel*, versão 1.0.17.

II. RESULTADOS E DISCUSSÕES

Nesta seção encontram-se os principais resultados de cada abordagem de modelo utilizada. Na Tabela IV estão descritos os valores referentes aos melhores ajustes de cada modelo, sendo possível observar também que tanto as medidas de acurácia quanto de sensibilidade ficaram muito próximas de 100%, principalmente por haver um desbalanceamento da base, visto que mais de 98% dos registros encontram-se apenas em uma classe.

TABELA IV. RESULTADOS DOS MODELOS AJUSTADOS

Modelo	Hiperparâmetros	Acc	Sens	Spec
RNA	size=1 decay=0.5	99,54%	99,99%	68,14%
SVM	sigma=0.05 C=2	99,52%	99,99%	67,26%
RF	mtry=3, ntree=500	**99,58%**	99,96%	**72,57%**
RL		99,52%	99,99%	67,26%

A. *Medidas de qualidade dos modelos*

No que diz respeito às métricas de qualidade dos ajustes analisados, nota-se que a acurácia ficou muito próxima entre todos eles, sendo o modelo RF o que melhor classificou, de modo geral, os estoques com relação à sua veracidade ou não. Com uma acurácia de 99,58%, o modelo em questão foi apenas 0,04% superior ao segundo melhor ajuste, obtido pelo modelo RL.

Com relação à sensibilidade, como se baseia na classificação correta dos estoques sem problemas de veracidade, ou seja, a maioria da base, com exceção do modelo RF, todos os modelos obtiveram valores próximos de 99,99%.

Por fim, a métrica em que mais foi observada variabilidade foi na especificidade. Novamente, o modelo que melhor se ajustou foi o RF, com 72,57%, seguido pelo modelo RNA, com 68,14%.

B. *Matrizes de Confusão*

As matrizes que deram origem aos resultados mencionados anteriormente encontram-se a seguir, nas Tabelas V, VI, VII e VIII. Nelas, o "Não" indica que a ordem de compra não apresentou problemas com relação à veracidade da informação de estoque fornecida pelo lojista e, consequentemente, o "Sim" implica em problemas com relação ao estoque.

TABELA V. MATRIZ DE CONFUSÃO DO MODELO RNA

		Realizado	
		Não	**Sim**
Predito	**Não**	7.886	38
	Sim	1	75

TABELA VI. MATRIZ DE CONFUSÃO DO MODELO SVM

		Realizado	
		Não	**Sim**
Predito	**Não**	7.886	37
	Sim	1	76

TABELA VII. MATRIZ DE CONFUSÃO DO MODELO RF

		Realizado	
		Não	**Sim**
Predito	**Não**	7.884	31
	Sim	3	82

TABELA VIII. MATRIZ DE CONFUSÃO DO MODELO RL

		Realizado	
		Não	**Sim**
Predito	**Não**	7.886	37
	Sim	1	76

C. *Discussões*

Este estudo teve por objetivo fazer uma comparação com relação à acurácia de técnicas comumente utilizadas na classificação binária, aplicadas a um problema real de uma empresa do setor de *e-commerce*. No caso, as técnicas utilizadas foram redes neurais artificiais, *support vector machine*, *random forest* e regressão logística.

Entretanto, como os resultados foram muito próximos de 100%, principalmente devido ao desbalanceamento entre as duas classes presentes na base, mesmo com um modelo se ajustando melhor que os outros, a variação foi mínima. Como o estudo não utilizou técnicas de tratamento para o desbalanceamento, este fato acaba impedindo a tirada de conclusões mais assertivas, além de não existir muitos casos semelhantes na literatura por se tratar de um problema específico da empresa em questão.

Visando amenizar os problemas de análise encontrados neste estudo, algumas sugestões poderiam ser testadas em análises futuras dessa base de dados, principalmente no que diz respeito ao desbalanceamento da base, como técnicas de estimação baseadas em *bootstrap*, utilizadas, por exemplo, no estudo de Ji-Hyun Kim [6] ou utilização de algoritmos mais sofisticados, como *transformers*, utilizado no experimento de aprendizado de linguagem natural em [5].

REFERÊNCIAS

[1] BREIMAN. Leo. "Random forests, " Mach Learn, vol. 45, 2001, p.5–32. DOI: 10.1023/A:1010933404324.

[2] CASTRO. Cristiano Leite de, BRAGA. Antônio Pádua. "Aprendizado supervisionado com conjuntos de dados desbalanceados," Controle & automação: revista da Sociedade Brasileira de Automática, vol.22, 2011, p.441-466. DOI: 10.1590/S0103-17592011000500002.

[3] CORTES. Corinna, VAPNIK. Vladimir. "Support-vector networks," Mach Learn, vol. 20, 1995, p.273–297. DOI: 10.1007/BF00994018.

[4] GUIMARÃES, M. "Exames de laboratório: sensibilidade, especificidade, valor preditivo positivo," Revista da Sociedade Brasileira de Medicina Tropical, vol. 18, 1985, p.117-120.

[5] KARITA, Shigeki, SOPLIN, Nelson Enrique Yalta, WATANABE, Shinji, DELCROIX, Marc, OGAWA, Atsunori, NAKATANI, Tomoshiro. “Improving Transformer-based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration,” Proc. Interspeech, 2019, p.1208-1412. DOI: 10.21437/Interspeech.2019-1938.

[6] KIM, Ji-Hyun. “Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap,” Computational Statistics and Data Analysis, vol. 53, 2009, p.3735-3745. DOI: 10.1016/j.csda.2009.04.009.

[7] KLEINBAUM, David G., KLEIN, Mitchel. “Logistic Regression: A Self-Learning Text,” Springer Science+Business Media, ed.3, 2010. DOI: 10.1007/978-1-4419-1742-3.

[8] KÖSTER, Andreas M., CAMPO, Jorge M. del, JANETZKO, Florian, GUTIERREZ, Bernardo Zuniga. “A MinMax self-consistent-field approach for auxiliary density functional theory,” The Journal of Chemical Physics, vol. 130, 2009. DOI: 10.1063/1.3080618.

[9] LINGAM. Yashoda Kiran. “The role of Artificial Intelligence (AI) in making accurate stock decisions in E-commerce industry,” International Journal of Advance Research, Ideas and Innovations in Technology, vol. 4, 2018, ISSN: 2454-132X.

[10] RAMEZAN, Christopher A., WARNER, Timothy A., MAXWELL, Aaron E. “Evaluation of Sampling and Cross-Validation Tuning Strategies for Regional-Scale Machine Learning Classification,” Remote Sens, vol. 11, 2019. DOI: 10.3390/rs11020185.

[11] ROSA. Paulo de Tarso Marques. “Modelos de 'credit scoring': regressão logística, 'chaid' e real,” Curso de Pós-graduação em Estatística, Instituto de Matemática e Estatística da Universidade de São Paulo, p.18. DOI: 10.11606/D.45.2000.tde-20210729-122749.

[12] SENAVIRATNA, N. A. M. R, COORAY, T. M. J. A. “Diagnosing Multicollinearity of Logistic Regression Model,” Asian Journal of Probability and Statistics, vol. 5, 2019, p.1-9. DOI: 10.9734/ajpas/2019/v5i230132.

[13] TABACHNICK, Barbara G., FIDELL, Linda S. “Using Multivariate Statistics,” Pearson College Div, ed. 5, 2006.