PREVISÃO EM UM SISTEMA DE VENDAS COMERCIAL A PARTIR DE ALGORITMOS DE INTELIGÊNCIA ARTIFICIAL

FORECAST IN A COMMERCIAL SALES SYSTEM BASED ON ARTIFICIAL INTELLIGENCE ALGORITHMS

[a]Neto, Henrique S. [a]Correa, Robson A. [a]Lochter, Johannes V.

seschin_eletronic@hotmail.com, robsonagapito@gmail.com, johannes.lochter@facens.br

[a]Centro Universitário Facens - Sorocaba, SP, Brasil

Submetido em: / /2022. Aprovado em: / /2022.

RESUMO

Atualmente, no período pós pandemia, propiciou que as lojas de colchões e estofados, pudessem aumentar sua visibilidade e vendas, através de ferramentas e comunicadores digitais. Devido a grande quantidade de informações registradas diariamente em um sistema de vendas de uma loja de colchões em Sorocaba-SP, o controle se tornou mais complexo, sendo que o sistema atual possui certas dificuldades para extrair informações relevantes e encontrar probabilidades de possíveis vendas futuras. Este trabalho propôs uma solução para extrair essas informações da base de dados da loja e tem como tarefa encontrar previsões das vendas futuras, através de algoritmos de inteligência artificial. Teve-se como objetivo neste trabalho, inserir técnicas de categorização de textos, pré-processamento de dados, redução de dimensionalidade e técnica para a diminuição do desbalanceamento das classes, para isto ocorrer, empregou-se o uso das técnicas Bag-of-Words, OrdinalEncoder, PCA e SMOTE. Para encontrar as previsões das vendas desse sistema, foram aplicados os métodos de classificação XGBoost, LightGBM, RandomForest e uma simples arquitetura de Rede Neural Artificial (RNA). Foram realizadas comparações sobre qual método de classificação obteve o melhor desempenho nas predições, avaliados através das métricas de validação F1-Score, Macro Average, Predict_Proba e Accuracy. Apesar do método XGBoost ter apresentado excelentes resultados quando seus dados foram treinados com a técnica do Bag-of-Words, ele se torna computacionalmente caro, quando treinado com grandes Datasets e também é mais lento do que o método LightGBM. Sugere-se para futuros trabalhos, usar uma base de dados maior e que esteja padronizada, para possibilitar a redução do desbalanceamento dos dados e também se sugere, buscar através da técnica GridSearch, por melhores hiperparâmetros nos métodos de classificação, no entanto, isto exigirá do uso de maiores recursos computacionais.

Palavras-chave: XGBoost. LightGBM. Saco-de-Palavras. Aprendizado de Máquina.

ABSTRACT

Currently, in the post-pandemic period, it has allowed mattress and upholstery stores to increase their visibility and sales, through digital tools and communicators. Due to the large amount of information recorded daily in a sales system of a mattress store in Sorocaba-SP, the control has become more complex and the current information system has difficulties to extract relevant information and find probabilities of possible future sales. This work proposed a solution to extract this information from the store's database and its task is to find forecasts of future sales, through artificial intelligence algorithms. The objective of this work was to insert text categorization techniques, data pre-processing, dimensionality reduction and technique to reduce class imbalance, for this to occur, the use of Bag-of-Words techniques was used. OrdinalEncoder, PCA and SMOTE. To find the sales forecasts of this system, XGBoost, LightGBM, RandomForest classification methods and a simple Artificial Neural Network (ANN) architecture were applied. Comparisons were made on which classification method had the best performance in the predictions, evaluated through the validation metrics F1-Score, Macro Average, Predict_Proba and Accuracy. Although the XGBoost method showed excellent results when its data were trained with the Bag-

of-Words technique, it becomes computationally expensive when trained with large Datasets and is also slower than the LightGBM method. For future works, it is suggested to use a larger Database and that it is standardized, to enable the reduction of data imbalance and it is also suggested to search through the GridSearch technique, for better hyperparameters in the classification methods, however, this will require the use of greater computational resources.

Keywords: XGBoost. LightGBM. Bag-of-Words. Machine Learning.

1 INTRODUÇÃO

O trabalho foi fundamentado em uma loja de colchões, da cidade de Sorocaba-SP. A base de dados utilizada, foi referente à quantidade de atendimentos realizados diariamente por esta determinada loja, sendo que os atendimentos aconteceram online ou presencialmente.

O mercado de colchoaria cresceu muito nos últimos anos e o perfil dos clientes mudou, principalmente com relação ao retorno de informações com uma maior agilidade. Pois antes, somente com o presencial, o atendimento era praticamente de imediato, e uma pessoa corria até 6 lojas para efetuar uma compra, hoje com a internet, o cliente busca por 6 lojas de forma on-line para filtrar os fornecedores e vai apenas em duas lojas para experimentar o produto e assim efetivamente comprar o mesmo. Deste modo, tornou-se muito importante a forma da comunicação virtual.

O primeiro objetivo deste trabalho, foi identificar se um novo cliente poderia ser um potencial prospect ou não, e caso não fosse possível identificar tal situação, orientar qual seria o caminho a percorrer, para que se possa obter esta informação em tempo de atendimento. Também se tem como objetivo captar informações estatísticas com relação aos dados coletados para geração de gráficos e dashboards, para tomadas de decisões gerenciais, e que apoiem nas estratégias de vendas e de marketing para os meses subsequentes.

Espera-se resolver este problema do trabalho, embasando-se em métodos e técnicas de algoritmos de inteligência artificial, utilizando-se das métricas avaliativas por predições, nas futuras vendas de uma loja de colchões, através da execução de métodos de classificação, através de árvores de decisão, algoritmos com *Gradient Boosting*, rede neural, processamento de linguagem natural e tratamento de dados. A seguir, na seção 2, serão abordadas as referências literárias citadas nesta contribuição acadêmica.

2 REFERÊNCIAS LITERÁRIAS

Abordando o assunto em vendas de produtos através de ferramentas digitais, segundo a SBVC (2020), a Sociedade Brasileira de Varejo e Consumo, a partir do início do ano de 2020, com a chegada da pandemia de Covid-19, os estabelecimentos comerciais se viram obrigados a reduzir suas atividades de maneira física, tendo que encontrar maneiras de seguir operando com as suas vendas pela internet, através de ferramentas tecnológicas, para possibilitar a sobrevivência do negócio.

De acordo com uma pesquisa feita pela Sociedade Brasileira de Varejo e Consumo, 61% dos entrevistados aumentaram a quantidade de compras feitas em e-commerce por causa do isolamento social. A migração para as compras digitais, aconteceu em diversas categorias, a praticidade parece ser a regra na hora de comprar online, onde 70% dos entrevistados preferem utilizar aplicativos e ferramentas de

comunicação das lojas em seus smartphones. Em seguida, o segundo dispositivo mais adotado pelos compradores são os computadores, que permitem a aquisição dos produtos através dos sites e aplicativos (SBVC, 2020).

Para a SBVC (2020), após a quarentena, 70% dos entrevistados pretendem continuar comprando mais em sites e aplicativos das lojas. Isso devido a uma satisfação de 78% nas experiências neste período de pandemia. O que significa dizer, que está havendo uma mudança real e significativa de comportamentos dos compradores brasileiros e as empresas que conseguirem se relacionar melhor com os clientes neste cenário digital, terão uma grande vantagem neste período pós-pandemia.

SBVC (2020) afirma que o Brasil é um dos países com o maior aumento de receita em e-commerce, devido ao novo coronavírus. Antes do período da pandemia, a percentagem de receita desta modalidade digital de vendas nas empresas era de apenas 42% e, com a mudança pós crise, aumentou as suas vendas através de meios digitais para 62%.

Seguindo as informações acima e para alcançar os objetivos deste trabalho, utilizou-se da avaliação dos resultados dos métodos de classificação. Assim, para poder avaliar os métodos, fundamentou-se nas pontuações das métricas *F1-Score*, *Macro Average, Accuracy* e *Predict_Proba.*

Segundo GÉRON (2019), muitas vezes é conveniente combinar a métrica *Precision* e *Recall* em uma única métrica, denominada métrica *F1-Score*, em particular, usa-se tal métrica quando necessita-se de uma maneira mais simples para comparar dois classificadores. A métrica *F1-Score* é a média harmônica da métrica *Precision* e *Recall*. Enquanto a média regular trata todos os valores de modo igual, a média harmônica tem muito mais peso para os valores baixos de classificação. Como resultado, o classificador só obterá uma pontuação alta de *F1-Score*, se tanto a métrica *Recall*, como a métrica *Precision* forem valores altos.

A F1-Score, favorece os classificadores que tem as métricas *Recall* e *Precision* semelhantes, deste modo, nem sempre é o que se pretende ter como resultados, ou seja, em determinados contextos se pretende focar apenas na métrica *Recall* e em outros contextos apenas focar na métrica *Precision*. Já para a métrica *Macro Average,* de um modo geral, calcula os valores para cada rótulo e encontra sua média não ponderada, não levando em conta o desbalanceamento dos rótulos do classificador (GÉRON, 2019).

Para GÉRON (2019), a métrica *Accuracy* é considerada umas das métricas mais simples e importantes de um modelo. Ela avalia simplesmente o percentual de acertos, ou seja, ela pode ser obtida através da razão entre a quantidade de acertos e o total das entradas.

GÉRON (2019) afirma que o método *Predict_Proba* retorna um *array* contendo uma linha por instância e uma coluna por classe, cada uma contendo a probabilidade de que a instância dada, pertença àquela determinada classe. Ou seja, os preditores dos métodos de classificação aplicados neste trabalho, podem estimar as probabilidades das classes, através do *Predict _Proba*.

Para abordagem dos métodos de classificação *Random Forest, XGBoost* e *LightGBM* usados neste trabalho acadêmico, foi essencial mencionar as referências sobre *Decision Trees*, em português, árvores de decisão.

O método utilizado por *Decision Trees*, em português, árvores de decisão, usam a estratégia da divisão, para resolver problemas de decisão, ou seja, um problema complexo é transformado em problemas mais simples, aos quais recursivamente é

aplicado a mesma estratégia. Deste modo, as soluções dos subproblemas podem ser combinadas, na forma de uma árvore, para produzir uma solução do problema complexo, de acordo com os autores (FACELI et al., 2011).

Para FACELI et al. (2011), a força desta teoria sobre árvores, vem da capacidade de dividir o espaço de instâncias em subespaços e cada subespaço é ajustado usando diferentes modelos. Essa é a ideia usada por trás dos algoritmos baseados em árvores de decisão. Os modelos em árvores, são nomeados como árvores de decisão no caso de problemas de classificação e árvores de regressão para os problemas de regressão.

Já para JOSHI et al. (2016), um outro conjunto de técnicas alternativas de *ensemble* de médias, usadas para um sistema de classificação, são as *Random Forests*, em português, florestas aleatórias.

De acordo com JOSHI et al. (2016), talvez a técnica de *ensemble* mais bem sucedida, usada por cientistas de dados competidores, são as *Random Forests.* Pois as *Random Forests,* desenvolvem conjuntos paralelos de classificadores, baseados em *Decision Trees* e ao inserir duas fontes principais de aleatoriedade ao construtor classificador da árvore, no final, a floresta acaba contendo diversas árvores de decisão.

O algoritmo *Random Forest*, introduz uma aleatoriedade extra ao processar suas árvores, em vez de procurar o melhor recurso ao dividir um nó, ele procura um melhor recurso entre um subconjunto aleatório de recursos. Isso resulta em uma maior diversidade de árvores processadas, trocando um valor anterior de variância maior, por uma variância com o valor menor, geralmente produzindo um modelo com performance melhor (GÉRON, 2019).

Para abordar os algoritmos de classificação *XgBoost e LightGbm*, será primordial citar o método *Boosting* e *Gradient Boost*. De acordo com JOSHI et al. (2016), uma outra forma alternativa de criação de *Ensemble* é construir modelos com *Boosting*, em português, modelos com “reforço”. Esses modelos, são caracterizados por usar vários modelos sequenciais em seu construtor, para poderem aumentar ou melhorar iterativamente o desempenho do *Ensemble*.

Modelos que usam *Boosting,* usam um nível de aprendizagem relativamente fraco e a cada iteração, um novo aprendizado é treinado em um conjunto de dados ajustados. Deste modo, ao longo de várias iterações, o *Ensemble* é estendido com uma nova árvore, cujo a qual foi a responsável por otimizar o desempenho da pontuação do modelo, em cada iteração (JOSHI et al., 2016).

GÉRON (2019) afirmou que, um outro algoritmo de *Boosting* muito usado popularmente é o *Gradient Boosting*. Assegurou que os modelos que usam o *Gradient Boosting* em seu construtor*,* funcionam adicionando preditores sequencialmente a um *Ensemble*, ou seja, a cada nova iteração, o modelo corrige o seu preditor anterior. Porém, em vez do algoritmo ajustar os pesos das instâncias a cada nova iteração do modelo, o *Gradient Boosting* tentará ajustar o novo preditor aos resíduos dos erros gerados pelo seu preditor anterior.

De acordo com JOSHI et al. (2016), em meados de 2015, surgiu o *XGBoost*, um novo algoritmo para resolver problemas estruturados de aprendizado de máquina, onde foi inserido na comunidade competitiva de ciência de dados. O *XGBoost*, *Extreme Gradient Boosting* é um algoritmo muito bem desenvolvido, por ser uma biblioteca de alto desempenho e fornece um algoritmo de aumento generalizado. A cada iteração o *XGBoost*, busca melhorar o desempenho do conjunto de modelos existentes, reduzindo os resíduos (as diferenças entre os *targets* e as previsões dos rótulos).

Os autores JOSHI et al. (2016) afirmam que, o *XGBoos*t é uma biblioteca escalável, com aumento de gradiente, disponível para Python, R, Java Scala e C++, pode ser usado tanto em máquinas singelas, como em clusters, tais como o Hadoop e Spark. Uma das melhorias mais importantes do *XGBoost* sobre o *(GBM) Gradient Boost Machine* é a capacidade de gerenciar dados esparsos. O método *XGBoost* aceita automaticamente dados esparsos como entrada, sem armazenar valores nulos numa memória.

Já SARKAR *et al*. (2018*)* complementam que, uma outra maneira de utilizar o método *Gradient Boosting* baseado em *Ensemble* e muito usado popularmente, aconselha-se a aplicar o modelo *XGBoost,* que significa *Extreme Gradient Boosting,* uma variante do modelo *Gradient Boost Machine (GMB)*. O modelo *XGBoost* é extremamente popular na comunidade de *Data Science*, devido ao seu desempenho superior em inúmeras competições e desafios de *Data Science*, especialmente no Kaggle.

O algoritmo *Gradient Boost Machine* é usado para máquinas de aprendizado supervisionado e apresenta um conjunto de aprendizado fraco. As implementações mais utilizadas com as técnicas de *Gradient Boost Machine (GMB)* são com os modelos *XGBoost e LightGBM*, segundo (Ke et al., 2017).

Para os autores KE et al. (2017), a melhor maneira de aproveitar todo o potencial do *LightGBM* é primordial ajustar corretamente a grande quantidade de hiperparâmetros disponíveis no *LightGBM*. A biblioteca *LightGBM,* tem um número muito alto de hiperparâmetros para serem otimizados, por isso é sempre importante pesquisar e testar os impactos que os hiperparâmetros podem trazer para os diferentes conjuntos de dados do *Dataset*.

O *LightGBM* é um modelo com estrutura de aumento de gradiente, que usa algoritmos de aprendizado baseados em árvore, o mesmo foi projetado para ser distribuído e eficiente, indicado para implementações que sejam altamente escaláveis, flexíveis e que precisam fazer uso de um sistema bem distribuído, para reduzir o tempo de treinamento. Permitindo assim que o modelo seja mais rápido durante o treinamento, menor uso de memória, melhor precisão e capaz de lidar com dados em grande escala, de acordo com (LightGBM's documentation, 2022).

Outro método de classificação utilizado neste trabalho, foi a implantação de uma rede neural artificial (RNAs), em inglês, *Artificial Neural Networks* (ANNs). A seguir, os autores FACELI et al. (2011), CHOLLET (2018) *e* GÉRON (2019), explicam respectivamente os fundamentos e anatomia das RNAs.

Para FACELI et al. (2011), as redes neurais artificiais, são sistemas computacionais distribuídos em unidades de processamentos simples, porém densamente interconectadas. Essas unidades, denominadas neurônios artificiais, computam funções matemáticas. As unidades estão dispostas em uma ou mais camadas e interconectadas por um grande número de conexões, geralmente unidirecionais. Na maioria das arquiteturas neurais, essas conexões, que simulam as sinapses biológicas, possuem pesos associados, que ponderam a entrada recebida por cada neurônio da rede neural, esses pesos podem assumir valores positivos ou negativos, dependendo do comportamento da conexão, podendo ser excitatório ou inibitório. Desta forma, os pesos têm seus valores reajustados em função do processo de aprendizado e codificam o conhecimento adquirido pela rede.

Uma RNA, é caracterizada por dois aspectos básicos: a arquitetura e o aprendizado. Enquanto a arquitetura está relacionada ao número de unidades de

processamento e a forma como os neurônios estão conectados, já o aprendizado está relacionado às regras utilizadas para o ajuste dos pesos da rede neural e qual a informação que será utilizada pela rede, segundo (FACELI et al., 2011).

CHOLLET (2018) explica que o treinamento de uma rede neural, gira em torno da seguinte anatomia:

- *Layers* (camadas), que são combinadas em uma rede ou modelo.
- *Inputs* (entradas) e os dados alvos, que são os *Targets*.
- *Loss Function* (função de perda), que define o sinal de *Feedback*.
- *Optimizer* (otimizador), que determina como o aprendizado prossegue na rede.

GÉRON (2019), menciona que as RNAs estão no centro do Deep Learning, elas são poderosas, versáteis e escaláveis. Tornando-as ideais para lidar com problemas extremamente grandes e complexos de *Machine Learning,* normalmente usados para resolver problemas de classificação e regressão.

Existe uma grande quantidade de dados disponíveis para treinar as redes neurais e as RNAs frequentemente superam todas as outras técnicas de Machine Learning, quando usadas para resolver problemas complexos, afirma (GÉRON, 2019). Com a tarefa de unir e extrair a grande quantidade de informações contidas em algumas colunas do *Dataset* deste trabalho, teve-se o intuito de criar vetores das palavras mais representativas dessas colunas, cujo as quais continham as maiores quantidades de textos, deste modo, utilizou-se de técnicas convencionais para fazer a categorização de texto, como a representação distributiva *Bag-of-words*, implantando assim a vetorização e a normalização nos conjuntos de dados do *Dataset*.

Segundo MOHAMMAD et al. (2020) ao contrário das palavras, não é viável pré-treinar *embeddings* para todas as sequências de palavras contidas em (frases, sentenças e textos), em uma linguagem natural, dado que o número de sequências possíveis neste caso pode ser infinito. Por isso, a representação para uma sequência de textos mais longos é geralmente computada com uma combinação das representações de suas palavras.

Os autores MOHAMMAD et al. (2020), explicam que a estratégia de combinação mais trivial neste caso é o modelo chamado de "*Saco de Palavras*", em inglês *Bag-of-words (BoW).* A representação para a sequência é obtida através da média das representações de suas palavras individuais. Em outras palavras, o *BoW* é um conjunto de palavras representado no seu ponto central do espaço vetorial. Uma importante questão com a representação do *BoW* é que todas as palavras desempenham um papel igual na representação final da sequência. No entanto, é natural esperar que algumas palavras da sequência possam ser mais centrais na sua semântica. Por isso, existem algumas variações do (BoW) que atribuem pesos às palavras, durante as combinações geradas.

Além disso, há uma outra questão em utilizar o (BoW*)* na representação de textos, que ainda permanece sem endereçamento. Essas representações chamam-se "*Saco de Palavras*" porque ignoram a ordem das palavras ao se combinarem, o que pode ser crucial em termos de semântica, segundo (MOHAMMAD et al., 2020).

De acordo com BENGFORT et al. (2016), os algoritmos de aprendizado de máquina, operam em um espaço de recursos numéricos, esperando a entrada como uma matriz bidimensional, onde as linhas são instâncias e as colunas são recursos. Respectivamente para realizar aprendizado de máquina em textos, precisa-se

transformar as instâncias e os documentos em representações vetoriais, para que seja possível aplicar o aprendizado de máquina numérico. O processo de codificação de documentos em um espaço de recurso numérico é chamado de Vetorização e é um passo essencial para a análise da linguagem.

A codificação mais simples no espaço semântico é o modelo do *Bag-of-Words* (BoW), deste modo, entende-se que o significado e a similaridade deste modelo, estão codificados em vocabulários específicos, usados para cada tipo de documento, explicou os autores (BENGFORT et al., 2016).

No que se refere a Vetorização, os autores BENGFORT et al. (2016) concluíram que o modelo mais simples de codificação vetorial é simplesmente preencher o vetor com a frequência que cada palavra aparece no documento, neste caso será um vetor de contagem de palavras. Nesse esquema de codificação, cada documento é representado como um multiconjunto que compõem o documento e o valor para cada posição de palavra no vetor é o número de vezes que a palavra aparece.

Com o objetivo de usar uma técnica para reduzir o tamanho da dimensão de um conjunto de dados do DataSet deste presente trabalho, preservando de certa forma suas propriedades, utilizou-se da técnica do *PCA.* A seguir, explica-se a técnica do *PCA - Principal Component Analysis.*

O autor NELLI (2015), menciona que quanto maior for o número de dimensões de um conjunto de dados, maior será o problema para se resolver, pensando nisso, desenvolveu-se uma técnica especial para reduzir as dimensões de um determinado conjunto de dados, chamada *Análise de Componentes Principais - (PCA)*. Esta técnica, permite reduzir o número de dimensões de um conjunto de dados, mantendo todas as informações principais da caracterização e as novas dimensões geradas, são chamadas de componentes principais.

Já para o autor GÉRON (2019) a Análise de Componentes Principais - (*PCA*), é de longe a técnica de algoritmo mais popular utilizada na redução de dimensionalidade, primeiro ela identifica o hiperplano mais próximo dos dados e, em seguida, ele projeta os dados nele. A técnica do *PCA*, identifica o eixo que responde pela maior quantidade de variação no conjunto de treinamento.

A seguir, na seção 3, serão descritos os materiais e os métodos utilizados neste trabalho acadêmico. Detalhando e citando a base de dados, as técnicas de pré processamento e tratamento de dados, os parâmetros utilizados nos modelos de classificação e as métricas usadas para a validação da solução.

3 MATERIAIS E MÉTODOS

A base de dados foi gerada a partir de um banco de dados, que foi obtido durante os atendimentos aos clientes, com informações registradas no dia a dia pelos vendedores da loja, tanto presencialmente com o cliente no interior da loja, como por meio de aplicativos de conversação. Como solução, foi proposto nesse experimento, utilizar-se da categorização de texto através da técnica de *Bag-of-Words*, da implantação de técnicas de processamentos para reduzir as dimensões dos conjuntos de dados, de técnicas de balanceamento e do uso de quatro métodos de classificação. Os resultados dos modelos de classificação foram avaliados através das métricas *F1-Score, Macro Average, Accuracy e Predict_Proba*. Também como um método deste experimento, foi

comparado os resultados dos métodos de classificação, treinando os modelos com e sem os dados convertidos pela técnica do *Bag-of-Words*.

Os detalhes da base de dados, da técnica de categorização de texto, da redução de dimensionalidade, do pré-processamento de dados, dos métodos de classificação e das métricas de validação, serão apresentados a seguir.

3.1 Base de dados

De acordo com VANDERPLAS (2016), um conjunto de dados podem vir de uma ampla gama de fontes e formatos, podendo ser coleções de informações, imagens, vídeos, sons e medições numéricas. Apesar dessa grande heterogeneidade de dados, é essencial usar a ferramenta *Numpy*, para pensar que todos os dados podem ser transformados em matrizes numéricas.

A biblioteca *Pandas*, baseia-se na estrutura de matrizes numéricas *NumPy* e fornece grande eficiência nas tarefas referentes ao tratamento de dados. *Pandas* é um dos pacotes mais recentes construídos em cima do *NumPy* e fornece uma implementação eficiente em um *DataFrame*. Dataframes são essencialmente arrays multidimensionais, com rótulos anexados em suas linhas e colunas, muitas vezes com dados ausentes (VANDERPLAS, 2016).

A base de dados deste trabalho, foi concedida por uma loja de colchões e estofados, pertencente a uma rede de franquias, situada na cidade de Sorocaba-SP. Esta base contém informações de clientes que visitaram a loja presencialmente ou através de aplicativos de comunicação. Todas essas informações foram trazidas para o *Dataset* usado neste trabalho. Devido ao grande número de informações registradas pelos vendedores da loja, foi necessário tratar a base de dados antes de realizar os treinamentos, usando das ferramentas para a ciência de dados, dentre elas, as bibliotecas *Pandas, NumPy* e *Scikit-Learn*, da linguagem *Python*.

Durante a limpeza e tratamento da base inicial de dados, foram utilizadas as ferramentas da biblioteca *Pandas*, para poder renomear as colunas, para transformar os dados nulos em informações coerentes, para substituir, remover e corrigir os erros de acentuação nos textos. Tais limpezas e tratamentos na base de dados, foram possíveis a partir do uso das ferramentas: *pandas.to_datetime*, *Dataframe.append, Dataframe.drop, Dataframe.rename, Dataframe.fillna, Dataframe.transpose* e *Dataframe.replace*, da biblioteca Pandas.

Devido a grande quantidade de produtos divididos em três colunas na base de dados, foi necessário criar uma coluna para cada produto, para isto, foi essencial concatenar e tratar as colunas, usando as ferramentas da biblioteca *NumPy*. Após a limpeza e tratamento de dados com as ferramentas das bibliotecas *Pandas* e *NumPy*, a base de dados inicial passou a conter 50 colunas de rótulos e 785 linhas de registros. Para o treinamento e validação dos modelos de classificação, a base de dados foi dividida em parcelas de treino e teste, contendo 70% dos dados para a parcela de teste e 30% dos dados para a parcela de treino.

3.2 Representação de Texto, Técnicas de Pré Processamento e Balanceamento

Bag-of-Words: Quando lidamos com documentos de textos que contém muitas palavras, precisa-se convertê-las em algum tipo de representação numérica. A razão disso é torná-las utilizáveis para o algoritmo de aprendizado de máquina. Esses algoritmos precisam de dados numéricos para que possam analisá-los e gerar resultados mais significativos. É onde a representação de texto por B*ag-of-Words* entra em cena, tratando-se basicamente de um modelo que aprende um vocabulário de todas as palavras contidas no conjunto de texto, depois disso modela cada conjunto de texto em um histograma, contendo todas as palavras daquele conjunto de texto, conforme foi proposto por (JOSHI et al., 2016).

Com o intuito de criar vetores das palavras mais representativas em duas colunas do *Dataset*, foi aplicado a técnica *Bag-of-Words* nas colunas *df['melhorias']* e *df['observacao']* do *Dataset*. Para isto ocorrer, usou-se da ferramenta *CountVectorizer* da biblioteca *Scikit-Learn* nas colunas *df['melhorias']* e *df['observacao']* do *Dataset*, cujo a qual converteu os textos das colunas em uma matriz de contagem, produzindo uma representação esparsa das contagens. Posteriormente a aplicação do *CountVectorizer*, utilizou-se do *fit_transform* nas duas colunas, para poder realizar o aprendizado dos vocabulários.

As colunas *df['melhorias'] e df['observacao']*, continham uma grande quantidade de texto, provenientes das descrições feitas pelos vendedores da loja durante os registros dos clientes. Contendo informações do que podia ser melhorado durante o atendimento ao cliente. Deste modo, essas duas colunas englobavam dezenas de palavras no *Dataset*, ficando inviável o treinamento dessas duas colunas. Portanto, foi construído a vetorização numérica e aplicado a técnica de *Bag-Of-Words* nas duas colunas que continham as maiores quantidades de textos no *Datase*t. Esse modelo foi executado de acordo com os autores (JOSHI et al., 2016).

PCA: A Análise de Componentes Principais *(PCA),* é uma técnica de redução de dimensionalidade muito usada em problemas de aprendizado de máquinas. Quando lidamos com recursos com grandes dimensionalidades, treinar um modelo de aprendizado se torna significativamente caro. Portanto tem-se a necessidade de reduzir a dimensionalidade dos dados antes de poder treinar um sistema, conforme (JOSHI et al*.,* 2016).

Após a execução da técnica *Bag-of-Words*, as duas colunas passaram a ter muitos rótulos, devido a grande quantidade de palavras vetorizadas durante o procedimento de *CountVectorizer* do *Bag-of-Words*. Deste modo, foi essencial aplicar a técnica de redução de dimensionalidade nas duas colunas geradas pelo *Bag-of-Words*. O número de componentes usados durante a aplicação da técnica de *PCA* nos *Bag-of-Words*, foi de 30 componentes. Para isto ocorrer na prática, foi necessário fazer a importação do método PCA, da biblioteca *Scikit-Learn* e aplicar o método nos dois *Bag-of-Words* que foram gerados. A aplicação da técnica de *PCA* foi essencial para evitar o overfitting dos modelos, a base de dados apresentava desbalanceamento antes da aplicação da técnica do *PCA*. Após a aplicação da técnica do *PCA*, o *bag_of_words_melhorias e o bag_of_words_observacao,* passaram *a* ter apenas 30 colunas rotuladas e 785 linhas de registros. Esta técnica do *PCA*, foi executada de acordo com a definição dos autores (JOSHI et al., 2016).

OrdinalEncoder: A maioria dos algoritmos de aprendizado de máquina, preferem trabalhar com números, de qualquer natureza que seja, então, sugere-se converter as categorias contendo textos em números. Para essa transformação ocorrer, pode-se usar a técnica de pré-processamento *OrdinalEncoder* da biblioteca *Scikit-Learning*, afirma o autor (GÉRON, 2019).

GÉRON (2019) também afirma, que no *OrdinalEncoder* os recursos são convertidos em números inteiros ordinais, resultando em uma única coluna de números inteiros (0 a n_categorias - 1). Após a aplicação da técnica de *Bag-of-Words* e o *PCA*, o *Dataset* final passou a possuir algumas colunas contendo informações em textos e outras colunas contendo informações com números. Deste modo, partiu-se para aplicação do pré processamento *OrdinalEncoder* da biblioteca *Scikit-Learn* nas parcelas *[X_train] e [X_test]* do *Dataset*, conforme proposto pelo autor (GÉRON,2019).

De acordo com GÉRON (2019), quando o parâmetro usado no *OrdinalEncoder* for setado *handle_unknow: "use-encoded_value",* esse parâmetro é obrigatório e definirá o valor decodificado das categorias desconhecidas. O valor codificado das categorias desconhecidas, será definido pelo valor fornecido no parâmetro *unknown_value*, neste caso = -1. Após a aplicação da técnica de pré-processamento *OrdinalEncoder* nas parcelas de treino e teste do *Dataset*, os conjuntos de dados de todo o *Dataset* passou a conter apenas números. Possibilitando assim avançar para as próximas etapas que antecedem os treinamentos dos modelos de classificação.

SMOTE: Para MOLIN (2019), *Synthetic Minority Oversampling Technique - SMOTE*, em português Técnica de Sobre Amostragem Minoritária Sintética, descreve que para conjunto de dados menores, não é benéfico subamostrar os dados.

Em vez disso, para conjuntos de dados menores, vale-se aplicar a técnica de sobre amostragem nas classes minoritárias para reduzir o desbalanceamento dos conjuntos de dados e amenizar o overfitting durante os treinamentos, deste modo, em vez de realizar uma sobre amostragem aleatória nos conjuntos de dados, faz-se o uso da técnica de *SMOTE*, para gerar novas classes (sintéticas) de alta qualidade. A técnica de *SMOTE,* foi originada a partir do artigo publicado dos autores (CHAWLA et al*.,* 2002).

Foi necessário aplicar a técnica de balanceamento *SMOTE* nos conjuntos de treino do *Dataset* final, com o objetivo de reduzir o overfitting durante os treinamentos dos modelos classificadores, devido ao fato da base de dados da franquia de colchões ser consideravelmente pequena e com muitos rótulos, causando assim um desbalanceamento nos dados, afetando os modelos durante os treinamentos. A execução da técnica *SMOTE*, foi aplicada nas parcelas de treinamento, *[X_train] e [y_train]* do *Dataset*, a partir da importação da biblioteca *Imblearn*, que é uma biblioteca de código aberto, licenciada pelo MIT e desenvolvida para lidar com classes desbalanceadas.

Para CHAWLA et al. (2002) os resultados mostraram que aplicar a técnica *SMOTE* em uma base de dados pequena, pode melhorar a precisão dos classificadores para uma classe minoritária. A combinação de *SMOTE* junto com uma subamostragem, tem melhor desempenho do que uma subamostragem simples.

3.3 Métodos de Classificação

Para esta contribuição acadêmica, teve-se como tarefa, aplicar quatro modelos eficientes para problemas de classificação, através de algoritmos de machine learning, para encontrar possíveis predições sobre um determinado cliente que realizou uma visita fisicamente na loja de colchões ou foi atendido através de meios de comunicação digital, com o intuito de prever se tal cliente irá ou não comprar os produtos da loja no futuro.

Após ter feito a busca pelos melhores parâmetros e executado os treinamentos dos modelos, realizou-se a comparação dos resultados de quatros métodos de classificação diferentes, comparando-os através das métricas de pontuação. No quadro 1, são apresentados os principais componentes empregados e os detalhes para permitir a reprodutibilidade dos experimentos de cada um dos quatro métodos de classificação executados. Todos os valores dos parâmetros aplicados nos modelos *XgBoost, LightGBM* e *Random Forest* foram seguindo as referências literárias mencionadas neste trabalho.

Os parâmetros mais importantes do método de classificação *XGBoost* usados neste experimento, foram retirados da referência: *Chapter 8, Ensemble Methods, Using XGBoost, p. 811,* mencionados pelos autores (JOSHI et al*.,* 2016).

Todos os parâmetros aplicados no método de classificação *LightGBM* usados neste experimento, foram retirados da documentação do *LightGBM,* são parâmetros padrões da biblioteca do *LightGBM*, podendo ser consultados em, (Welcome to LightGBM´s documentation, 2022)*.*

Para os parâmetros do método de classificação *Random Forest* usados neste trabalho, foram retirados de exemplos práticos, encontrados na referência: *Chapter 2, Constructing a Classifier, p. 77,* descritos pelos autores (JOSHI et al*.,* 2016).

Já para os parâmetros da arquitetura da rede neural artificial, foram aplicados intervalos de valores, variando entre uma faixa de valores da biblioteca do Keras, observando-se quais deles trariam as melhores pontuações na classificação do modelo.

Na Rede Neural Artificial, foi aplicada a normalização no conjunto *[X_train] e* para o conjunto de treinamento foram utilizadas 35 épocas, um split de validação de 0.2 e um tamanho de batch de 10.

A seguir no quadro 1, detalham-se os parâmetros dos métodos de classificação e a arquitetura do *Keras* que foi utilizada na Rede Neural Artificial (RNA).

Quadro 1 - Configuração dos Parâmetros dos Métodos e Arquitetura da Rede Neural

XGBoost - XGBClassifier
- n_estimators = 100; *(número de árvores possíveis na floresta)* - max_depth = 8; *(número de interações nas árvores)* - learning_rate = 0.1; *(taxa de aprendizado do modelo)* - subsample = 0.5; *(proporção de amostragem dos dados de treinamento)* - use_label_encoder = True; *(habilitar o encoder numérico para as classes)* - eval_metric = ‘mlogloss.’ *(métrica da função loss para as multiclasses)*

LightGBM - LGBMClassifier
- num_leaves = 31; *(número de folhas nas árvores)* - max_depth = -1; *(número de interações nas árvores)* - learning_rate = 0.1; *(taxa de aprendizado do modelo)* - n_estimators = 100; *(número de árvores possíveis na floresta)* - xgboost_dart_mode = True. *(usar o modo Dart do Xgboost, no modelo LightGBM)*
Random Forest - RandomForestClassifier
- n_estimators = 200; *(número de árvores possíveis na floresta)* - max_depth = 8; *(número de interações nas árvores)* - random_state = 7 *(aleatoriedade dos valores, podendo ser uma seed)*
RNA - Rede Neural Artificial - Keras.Sequential
- normalizer = preprocessing.Normalization(); - Camada Densa, com 35 neurônios e ativação por ReLU; - Camada Densa, com 1 neurônio e ativação por Sigmoid; - Otimizador = sgd; - Função de Custo = binary_crossentropy; - Métricas = Accuracy. Intervalo dos parâmetros testados durante o treinamento da RNA: - Camada Densa: [30 - 35] neurônios, o menor loss e a maior acurácia foi com [35]; - Camada Densa: [1 - 5] neurônios, o menor loss e a maior acurácia foi com [1]; - Número de Épocas: [30 - 50] épocas, o menor loss e a maior acurácia foi com [35]; - Tamanho do Batch: [5 - 30] batch, o menor loss e a maior acurácia foi com [10].

Fonte: Elaborado pelos autores.

3.4 Métricas

Para ser possível realizar a verificação da eficácia da previsão dos quatro modelos classificadores, tendo como *Targets* as saídas: [vendeu=1] e [vendeu=0], predizendo a venda de produtos de uma loja de colchões, foram utilizadas as métricas *F1-Score*, *Macro Average*, *Accuracy* e *Predict_Proba.*

A métrica *Macro Average,* foi escolhida devido ao fato que a base de dados deste trabalho, possui um desbalanceamento considerável, deste modo a *Macro Average* é a métrica de verificação mais adequada para casos de dados com desbalanceamento. Ao se resolver qualquer problema de machine learning, na maioria das vezes a escolha da medida de desempenho é a *Accuracy, F1- Score, Precision e Recall.* A métrica *Macro Average ou Macro F1-Score*, das médias, talvez seja a mais simples entre os vários métodos de médias existentes no *Classification Report*. A métrica *Macro Average* é calculada através da média aritmética, sobre todas as pontuações do *F1-Score*, obtidas por classe. Esse método, trata todas as classes da classificação de modo igual, independente dos seus valores.

Neste trabalho necessitou-se de uma métrica que requer uma otimização balanceada de *Precision e Recall*, então, utilizou-se da métrica *F1-Score,* que é uma

métrica eficiente, pois permite uma média harmônica entre *Precision e Recall*, ajudando a otimizar um classificador, resultando em uma precisão mais equilibrada para os métodos de classificação usados neste trabalho. A métrica *Precision*, também é conhecida como um valor preditivo positivo, pois é uma métrica que pode ser derivada da matriz de confusão. O *Recall,* também conhecido como sensibilidade ou taxa de acerto, é definido como o número de instâncias das classes positivas, que foram corretamente previstas. Conforme descrevem as expressões abaixo:

- TP: *True positive* (verdadeiro positivo), quantidade de dados rotulados como positivos que foram classificados corretamente.
- TN: *True negative* (verdadeiro negativo), quantidade de dados rotulados como negativo e classificados corretamente.
- FP: *False positive* (falso positivo), quantidade de dados rotulados como negativos, mas foram classificados como positivos.
- FN: *False negative* (falso negativo), quantidade de dados classificados como negativos, mas estavam rotulados como positivos.

A métrica *F1-Score* é definida entre a *Precision (P) e Recall (R)*, sendo a *Precision* definida por P = TP / (TP + FP) e a *Recall* expressa como R = TP / (TP + FN). Assim, a *F1-Score* é definida pela expressão final: 2PR / (P+R)

De acordo com GÉRON (2019) a métrica *F1-Score* é uma métrica adequada para validar casos onde os conjuntos de dados estão com classes desbalanceadas.

E para se obter uma outra pontuação de avaliação neste experimento, usou-se da métrica *Accuracy,* do *Classification Report,* cujo a sua expressão é dada por: Accuracy = (TP + TN) / (TP + TN + FP +FN). Porém, conforme afirma o autor GÉRON (2019), a métrica *Accuracy* por si só, não informa todos os valores corretamente, de quando está se trabalhando com um conjunto de dados que possuem classes desbalanceadas, deste modo, a metodologia de validação deste trabalho foi propor paralelamente outras métricas do *Classification Report,* para problemas de dados desbalanceados.

Finalmente, o método *Predict_Proba* também foi aplicado neste experimento, com o intuito de informar quais seriam as probabilidades das predições dos métodos de classificação, em acertar ou errar uma futura venda, neste caso a predição foi feita sobre a classe [vendeu = 0] ou [vendeu = 1].

O método *Predict_Proba* é um método pertencente à biblioteca *Scikit-Learn* e foi aplicado nos métodos de classificação *XGBoost, LightGBM e Random Forest* deste trabalho.

A seguir na seção 4, estão detalhados os resultados encontrados neste experimento e as discussões sobre os métodos de classificação, que foram utilizados para encontrar as previsões da classe [‘vendeu’].

4 RESULTADOS E DISCUSSÃO

Os resultados das previsões deste presente trabalho, encontrados a partir dos quatro métodos de classificação, estão apresentados através das métricas *F1- Score*, *Macro Average, Predict_Proba* e *Accuracy*, conforme detalham os quadros 2 e 3. A classe que foi utilizada como *Target* para a predição dos modelos, foi a classe [vendeu=0] e [vendeu=1], onde 0 significa que não venderá um produto no futuro e 1 significa que o produto será vendido. Os valores em negrito, representam as maiores pontuações das métricas obtidas, dentre os quatro métodos de classificação que foram comparados nos quadros abaixo.

Para efeitos de conclusão, comparou-se quais os modelos que obtiveram os melhores resultados, aplicando ou não a técnica do *Bag-of-Words* no conjunto de dados de treinamento. Para posteriormente concluir, se a aplicação da técnica de *Bag-of-Words* nos métodos de classificação, foi relevante ou não, para os resultados das predições deste experimento.

A seguir, no quadro 2, detalham os resultados dos métodos de classificação, usando em seus treinamentos os dados processados com a técnica *Bag-of-Words* e no quadro 3, detalham os resultados dos métodos de classificação sem a aplicação da técnica *Bag-of-Words* no treinamento dos modelos. No quadro 2 abaixo, demonstra que entre os quatro métodos de classificação comparados, aplicando a técnica de *Bag-of-Words no conjunto de treinamento*, os melhores resultados encontrados, foram obtidos através dos respectivos métodos de classificação: *XGBoost Classifier, LightGBM Classifier, Random Forest e Rede Neural Artificial.*

Quadro 2 - Resultados dos métodos de classificação, **com** a técnica *Bag-of-Words*

Métodos de Classificação	F1- Score vendeu=0	F1- Score vendeu=1	Macro Average F1-Score	Predict_Proba vendeu=0	Predict_Proba vendeu=1	Accuracy
XGBoost	**95%**	**60%**	**77%**	91.86%	**91,87%**	**90%**
LightGBM	**95%**	53%	74%	**99,34%**	88,77%	**90%**
Random Forest	93%	48%	71%	70,78%	74,41%	88%
Rede Neural	92%	47%	70%	-	-	86%

Fonte: Elaborado pelos autores.

Entre os 4 métodos comparados no quadro 2, que utilizaram dos dados do *Bag-of-Words* em seus treinamentos, os resultados foram semelhantes entre o melhor modelo *XGBoost* e o segundo melhor modelo *LightGBM*, com uma margem de apenas 3% para a métrica *Macro Average*.

Já para a métrica *F1-Score* na classe vendeu=0, a pontuação de 95% foi igual tanto para o *Xgboost* como para o *LightGBM,* com uma diferença de 7% para a métrica *F1-Score* na classe vendeu=1. Ambos os modelos obtiveram 90% de acurácia. Deste modo, comparando os modelos que utilizaram em seu treinamento os dados que passaram pela técnica do Bag-of Words, os dois melhores modelos que obtiveram as maiores pontuações nas métricas de validação foram o *XGBoost Classifier e LightGBM Classifier.*

Finalmente, para a métrica de probabilidade *Predict_Proba*, o modelo *LightGBM* obteve uma maior pontuação no *Predict_Proba* para a classe vendeu=0, com uma margem de 7% a mais do que a mesma métrica de probabilidade no *XGBoost*.

Em terceiro lugar na pontuação das métricas, ficou o modelo *Random Forest* e em quarto a Rede Neural Artificial, com uma margem de 3% apenas entre o primeiro e segundo lugar na métrica F1-Score, uma margem de 7% na Macro Average e uma pequena margem de 4% para a Accuracy. Ou seja, os modelos de classificação tiveram resultados semelhantes, com no máximo uma diferença de 10 pontos entre o primeiro e o último lugar nas comparações.

Apesar da pequena diferença entre os dois primeiros modelos, o modelo *XgBoost* foi o que obteve a maior pontuação dentre todas as métricas de validação. Resultando assim no melhor modelo de classificação para este determinado experimento, tendo em seu treinamento os dados processados pela adição da técnica do *Bag-of-Words*.

No quadro 3 abaixo, demonstra que entre os quatro métodos de classificação comparados, sem aplicar a técnica do *Bag-of-Words no treinamento*, os melhores resultados encontrados, foram obtidos através dos respectivos métodos de classificação: *XGBoost Classifier, Rede Neural Artificial, LightGBM Classifier* e *Random Forest.*

Quadro 3 - Resultados dos métodos de classificação, **sem** a técnica *Bag-of-Words*

Métodos de Classificação	F1- Score vendeu=0	F1- Score vendeu=1	Macro Average F1-Score	Predict_Proba vendeu=0	Predict_Proba vendeu=1	Accuracy
XGBoost	**95%**	58%	**76%**	99.64%	35,09%	**91%**
Rede Neural	94%	**59%**	**76%**	-	-	89%
LightGBM	94%	53%	74%	**99,99%**	35,72%	90%
Random Forest	93%	48%	70%	93,60%	47,08%	87%

Fonte: Elaborado pelos autores.

Entre os 4 métodos comparados no quadro 3, que não utilizaram dos dados do *Bag-of-Words* em seus treinamentos, os resultados foram semelhantes para o melhor modelo *XGBoost,* que também obteve o primeiro lugar na pontuação. Porém não treinando os modelos com a técnica do *Bag-of-Words*, o segundo lugar ficou com a Rede Neural, onde obteve um aumento de até 12% na métrica F1-Score, deste modo a Rede Neural foi da última posição no quadro 2 para a segunda posição no quadro 3, com apenas 1% de diferença em relação ao primeiro lugar, praticamente se igualando com o modelo *XGBoost*.

Em relação ao quadro 3, o modelo que mais obteve melhoras nas pontuações foi a Rede Neural, que certamente possibilitou um melhor aproveitamento no aprendizado da rede neural, quando foi treinada sem a técnica do *Bag-of-Words*, onde teve a adição de todas as frases das classes “melhorias” e “observação” do *Dataset,* incluídas em seu conjunto de treinamento, obtendo assim melhores resultados nas pontuações das métricas de validação.

Em terceiro e quarto lugar nas pontuações, ficaram os modelos *LightGBM* e *Random Forest,* que também tiveram diminuição de até 2% nas suas pontuações.

Deste modo, para os modelos *XGBoost, LightGBM* e *Random Forest*, que utilizam de árvores e *gradient boost* em sua estrutura, as pontuações de todas as métricas caíram

em relação ao quadro 3, com uma pequena diminuição de até 2% nas métricas *F1-Score, Macro Average* e *Accuracy*.

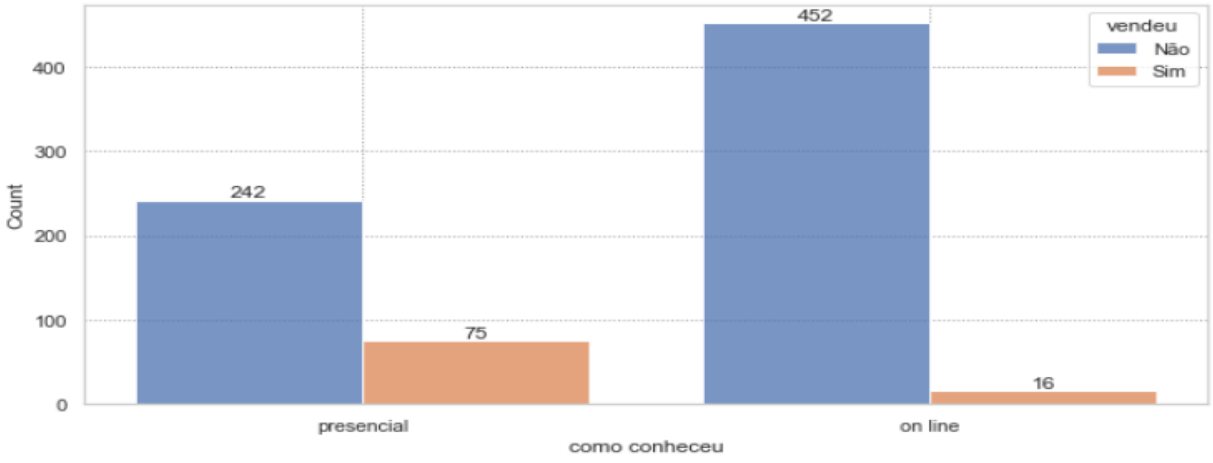
Finalmente para as pontuações de probabilidades do *Predict_Proba*, a diferença do quadro 2 para o quadro 3 foi significativa, pois os valores de todos os modelos diminuíram em até 57% para a m.étrica *Predict_Proba* na classe vendeu=1, para o modelo *XGBoost*.

Entende-se que a partir dos resultados encontrados e embasado na literatura, que foi relevante aplicar a técnica do *Bag-of-Words* nas classes “melhorias” e "observação" no conjunto de treinamento dos modelos, pois conforme os resultados do quadro 2, todos os métodos que possuem em sua estrutura o *gradient boost* e foram treinados com a técnica do *Bag-of-Words*, obtiveram os melhores resultados nas métricas de validação, com ganhos de até 2% em todas as principais métricas e ganhos significativos de até 57% nas probabilidades do *Predict_Proba*, quando comparados com o quadro 3, que não usaram da aplicação da técnica do *Bag-of-Words* nos dados de treinamentos dos modelos.

Além de predizer uma possível compra a partir de dados do DataSet, também foi possível identificar e gerar informações para a gestão da loja tomar algumas decisões importantes no dia a dia. A partir dos dados encontrados no *Dataset*, para efeitos de comparação de como o público conheceu a loja, foi gerado um gráfico, no qual demonstra essas estatísticas.

Essas estatísticas foram encontradas a partir da manipulação do *Dataset* deste trabalho, onde confirmou, que atualmente no período pós pandemia, a busca do público para a compra de produtos no mercado de colchões e estofados, tem crescido muito através das ferramentas digitais, dentre elas estão: *google, landing page, instagram, facebook, whatsapp* e telefone. Conforme a figura 1 abaixo:

Figura 1 - Primeiro contato com o cliente: Presencial x Online.



Fonte: Elaborado pelos autores.

Nesta figura, comparou-se o público presencial do público on-line, evidenciando que a busca por produtos através das ferramentas on-line, tem sido maior do que as visitas presencialmente na loja, ou seja, até o presente período de atividade da loja,

foram realizadas 317 visitas presenciais no interior da loja, contra 468 pessoas que conheceram a loja através das ferramentas digitais.

Tanto para o público que conheceu a loja presencialmente, como para os que conheceram a loja on-line, foram realizadas vendas. Enfim, a figura 1, concretiza o contexto de que as ferramentas digitais, têm sido um indispensável recurso que as lojas de colchões possuem para expandir sua visibilidade e aumentar suas vendas. A seguir na seção 5, são apresentadas as conclusões e as considerações finais desta proposta acadêmica.

5 CONCLUSÃO

Atualmente, no ano deste presente trabalho, no período pós pandemia, as procuras e vendas por produtos através de comunicadores e aplicativos digitais, aumentaram em todos os setores comerciais, inclusive em lojas de colchoarias e estofados, fazendo dessa prática uma ferramenta essencial para que os comércios pudessem flexibilizar as suas vendas, com muita praticidade e rapidez no atendimento ao público. Fundamentado nas referências literárias, este trabalho teve como proposta a previsão de futuras vendas em uma franquia de colchões e estofados na cidade de Sorocaba-SP, através de algoritmos de inteligência artificial, com técnicas de pré-processamento de dados e aplicando métodos de classificação nas predições.

Este trabalho acadêmico, analisou as diversas técnicas de pré-processamento de dados presentes na literatura, aplicando uma técnica de categorização de texto e usando técnicas para reduzir o desbalanceamento das classes do *Dataset,* com objetivos de melhorar o desbalanceamento dos dados e obter as melhores pontuações nas métricas de validação. Frequentemente usado na literatura, aplicou-se neste trabalho uma técnica de categorização de texto, denominada técnica do *Bag-of-Words.* Aplicando esta técnica nas duas classes do *Dataset* que continham maiores quantidade de textos, que foram as classes “observação” e “melhorias”.

Com a aplicação da técnica do *Bag-of-Words* nas classes “observação” e “melhorias” do *Dataset*, que continham uma grande quantidade de frases, os dados passaram a conter um grande número de classes, resultantes do procedimento do *Bag-of-Words*. Sendo essencial o estudo e aplicação de outras técnicas, antes de atribuir os conjuntos de dados para o treinamento dos métodos de classificação. Diante disso, embasado na literatura, conclui-se que após o uso do *Bag-of-Words*, foi indispensável a aplicação da técnica da Análise de Componentes Principais (*PCA),* para possibilitar a redução de dimensionalidade no *Dataset*, reduzindo assim a grande quantidade de classes do *Bag-of-Words* para apenas 30 categorias, resultando em um Dataset final com menores quantidades de classes e reduzindo o overfitting durante o treinamento da Rede Neural Artificial *(RNA)*.

O *Dataset* final deste trabalho, resultou na união de classes contendo informações em textos e números, por isso foi essencial ter usado o *Ordinal Encoder* no conjunto de dados*,* possibilitando a conversão de todos os textos das classes do *Dataset* para valores numéricos, sendo um procedimento relevante para regularizar o *Dataset* e permitir seguir corretamente com o treinamento dos modelos. Pois os modelos *XGBoost e LightGBM,* são métodos que aceitam apenas recursos numéricos em sua estrutura, isso significa que todas as características nominais do *Dataset* precisam ser transformadas em

características numéricas, por isso tornou-se indispensável o uso do *Ordinal Encoder* nessa proposta acadêmica.

Referente a técnica *Smote* (*Synthetic Minority Oversampling Technique*) em português, Técnica de Sobre Amostragem Minoritária Sintética, foi pertinente discutir que, ter aplicado esta técnica nos conjuntos de dados *[X_train]* e *[y_train]* foi significativa para esta proposta acadêmica, pois o autor MOLIN (2019), afirma que esta técnica apresenta melhores resultados em *Datasets* menores e permite reduzir o desbalanceamento nas classes minoritárias, neste caso, sendo compatível com o *Dataset* empregado neste trabalho, cujo o qual possui apenas 785 linhas de cadastros e um considerável número de classes, devido ao pouco tempo de funcionamento da loja de colchões e estofados na cidade de Sorocaba-SP. Deste modo, a execução da técnica *SMOTE* foi expressiva para a realização dessa proposta de previsão de vendas, pois os autores CHAWLA et al*.* (2002), citam que a técnica *SMOTE* quando aplicada a base de dados menores, pode melhorar a precisão dos classificadores, reduzindo o desbalanceamento entre as classes do *Dataset*. Finalmente, ter usado o *SMOTE*, possibilitou que os modelos pudessem ser treinados corretamente, com os conjuntos de dados balanceados e com menores margens de ocorrer um *overfitting* durante os seus treinamentos.

Alicerçado na literatura e nos resultados encontrados, conclui-se que é iminente antes de treinar os modelos, empregar as técnicas de pré-processamentos e desbalanceamento, quando se defrontar com um *Dataset* que foi identificado com algum tipo de desbalanceamento na base de dados. Na escolha dos métodos de classificação propostos neste trabalho, que foram os métodos *XGboost, LightGBM, Random Forest* e uma Rede Neural Artificial *(RNA)*, cujo os quais foram previamente encontrados na literatura e propostos para predizer a classe [vendeu=0] ou [vendeu=1], onde 0 significará uma previsão de que não acontecerá a venda e 1, a venda será realizada. O motivo da escolha desses métodos de classificação, foi embasado nas literaturas e competições realizadas globalmente entre os cientistas de dados, onde os métodos *XGboost, LightGBM* e árvores de decisão estão entre os métodos mais utilizados por esses profissionais.

Os algoritmos com reforço de gradiente oferecidos pelas comunidades e empresas, ganharam muita força nos últimos anos. Isso se deve principalmente à melhoria no desempenho oferecido pelas árvores de decisão em comparação a outros algoritmos de aprendizado de máquina. Enfim, dois dos métodos mais populares baseados em *Gradient Boosted Machines*, são o *XGboost* e *LightGBM.*

Entre esses métodos propostos de classificação, foi uma tarefa encontrar os parâmetros corretos para se obter os melhores treinamentos dos modelos, para isto, foi primordial ter pesquisado evidências de parâmetros adotados nas literaturas. Deste modo, encontrou-se parâmetros aplicados nos métodos *XGboost, LightGBM* e *Random Forest,* fazendo com que está busca na literatura, se tornasse uma indispensável ferramenta, para se obter a parametrização correta dos métodos e resultando em um bom desempenho nas previsões deste trabalho. Na conclusão comparativa dos métodos de classificação utilizados neste trabalho, foi notável a partir dos resultados encontrados que ao se utilizar dos métodos que possuem em sua estrutura o *Gradient Boosted Decision Trees* (GBDTs) e quando adicionados com a técnica do *Bag-of-Words* nas colunas [“melhorias”] e [“observação”] do *Dataset,* cujo as quais continham mais textos, a pontuação das métricas de validação foram superiores, quando comparadas as

pontuações das métricas dos mesmos modelos que foram treinados sem a técnica do *Bag-of-Words.*

Conclui-se que, ter aplicado a técnica do *Bag-of-Words* foi essencial para categorizar corretamente as classes que estavam em textos e possibilitando maiores pontuações nas métricas de validação. Além do aumento das pontuações nas previsões dos modelos, a técnica da categorização de textos por *Bag-of-Words* também foi relevante para realizar a transformação dos textos em valores numéricos, já que os métodos *XGboost* e *LightGBM* que possuem *Gradient Boosted Decision Trees* (GBDTs) em sua estrutura, aceitam apenas classes numéricas em seus treinamentos.

Diante dos resultados obtidos e das literaturas encontradas sobre os modelos *XGboost* e *LightGBM,* que possuem *Gradient Boosted Decision Trees* (GBDTs) em sua estrutura, conclui-se que foram esses os dois métodos de classificação que obtiveram os melhores resultados para esta proposta de trabalho, ficando respectivamente em primeiro e segundo lugar, dentre os quatro métodos comparados.

A diferença de pontuação nas métricas de validação entre o *XGBoost* em primeiro lugar e o *LightGBM* em segundo lugar foi de apenas 3%, deste modo, entende-se que ambos os algoritmos estabeleceram um alto padrão em termos de desempenho. A diferença fundamental entre os dois métodos, é que no *XGBoost* as árvores crescem em profundidade e no *LightGBM* as árvores crescem em folhas.

Para esta proposta de trabalho, diante de uma base de dados relativamente pequena, ter usado o método *XGBoost* ou o *LightGBM* não necessitou do uso de altos recursos computacionais durante os seus treinamentos. Tanto no desenvolvimento deste trabalho, como descrito na literatura, o tempo de treinamento do *XGBoost* foi maior do que o modelo *LightGBM*. Para bases de dados maiores o *XGBoost* levará mais tempo de treinamento do que o *LightGBM*. O método *XGboost,* necessita de muitos recursos computacionais quando usado para treinar uma grande quantidade de dados, enquanto o *LightGBM* é leve e pode ser treinado com um *hardware* modesto, que foi o caso deste trabalho.

Outra consideração importante, que foi notada durante essa pesquisa é que atualmente o método *XGBoost* é concretizado pelo alto número de usuários e comunidades pelo mundo, resultando em literaturas enriquecidas, na forma de livros, documentações, artigos publicados e resolução de problemas, enquanto o método *LightGBM* ainda não atingiu esse nível de documentação literária.

A partir deste trabalho, concluiu-se que, usar de métodos de classificação que possuem o *Gradient Boosted Decision Trees* (GBDTs) em sua estrutura e quando adicionadas corretamente as técnicas de pré-processamento para tratamento e desbalanceamento de dados, encontram-se consideráveis resultados nas métricas de validação. O fato de escolher entre adotar ao uso do método *XGBoost ou LightGBM* é uma escolha que deve ser analisada e decidida pelo usuário, visto nos resultados, que ambos os métodos possuem desempenhos praticamente iguais nas saídas. Por fim, o que deve ser levado em consideração na escolha dos métodos de classificação, são os recursos computacionais que poderão ser usados durante os treinamentos, sempre analisando a natureza e o tamanho dos dados.

Conclui-se neste trabalho, que com o banco de dados atual da loja de colchões, não é possível identificar com precisão qual a probabilidade de se prever uma possível venda dos produtos da loja. Pois a base possui dados não padronizados, para quando se efetuou uma venda, onde muitas vezes é apenas descrito brevemente pelos

vendedores como: “fechamento com vendas” e não uma informação correta dos métodos e técnicas que foram utilizadas naquela determinada venda. Como uma possível solução, seria utilizar as mensagens obtidas através das ferramentas online, tais como: *Whatsapp, Facebook e Instagram*, ao invés de se utilizar os campos “observação” e “melhorias”.

Além disso, para trabalhos futuros, com propostas de previsão em um sistema de vendas de uma loja de colchões e estofados, usufruindo também de uma base de dados genuína, sugere-se a busca por uma base de dados com registros superiores a 6 meses de funcionamento da loja, para se obter um *Dataset* com mais registros de vendas finalizadas, podendo assim resultar na eliminação do desbalanceamento entre classes e com maiores margens de acertos nas previsões.

Sugere-se também, após ter em mãos uma base de dados maior e padronizada, aplicar a técnica do *GridSearch,* para buscar os melhores parâmetros nos métodos de classificação utilizados neste trabalho, a fim de se obter melhores resultados nas métricas de validação. No entanto, para ser possível aplicar a técnica do *GridSearch* em todos os modelos e com uma base de dados maior, exigirá de maiores recursos computacionais e horas de treinamento. A partir dessas sugestões na base de dados e buscas por hiperparâmetros, será possível estabelecer quais serão os melhores métodos de classificação para se realizar uma previsão de vendas em uma loja de colchões e estofados.

REFERÊNCIAS BIBLIOGRÁFICAS

BENGFORT, B.; BILBRO, R.; OJEDA, T. (2016) Applied Text Analysis with Python: Enabling Language Aware Data Products with Machine Learning, First Edition. United States of America: O’Reilly Media

CHOLLET, F. (2018) Deep Learning with Python, First Edition. Shelter Island: Manning Publications

FACELI, K.; LORENA, C. A.; GAMA, J.; de CARVALHO, C. P. L. F. A. (2011) Inteligência Artificial: Uma Abordagem de Aprendizagem de Máquina, Primeira Edição. Rio de Janeiro: LTC.

GÉRON, A. (2019) Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems, Second Edition. Tokyo: O’Reilly Media.

JOSHI, P.; HEARTY, J.; SJARDIN,B.; MASSARON,L.; BOSCHETTI, A. (2016) Python: Real World Machine Learning, First Edition. Mumbai: Packt Publishing Ltd.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). Lightgbm: A highly efficient gradient boosting decision tree. Disponível em: http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf Acesso em: 12 julho. 2022.

MOHAMMAD, T.; JOSE, C. (2020) Embeddings in Natural Language Processing: Theory and Advances in Vector Representation of Meaning. Synthesis Lectures on Human Language Technologies, First Edition. Morgan & Claypool Publishers.

MOLIN, S. (2019) Hands-On Data Analysis with Pandas: Efficiently perform data collection, wrangling, analysis, and visualization using Python, First Edition. Birmingham. Mumbai. Packt

N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer (2002), SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, 321-357, 2002, Disponível em: https://arxiv.org/pdf/1106.1813.pdf Acesso em: 12 julho. 2022.

NELLI,F. (2015) Python Data Analytics: Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language, First Edition. New York: Apress

SARKAR, D.; BALI, R.; SHARMA, T. (2018) Practical Machine Learning with Python. A Problem - Solver´s Guide to Building Real-World Intelligent Systems, First Edition. Bangalore: Apress.

VANDERPLAS, J. (2016) Python Data Science Handbook : Essential Tools for Working with Data, First Edition. United States of America. O' Reilly Media.

XGBoost vs LightGBM: How Are They Different (2022). 10 fevereiro. 2022. Disponível em: https://neptune.ai/blog/xgboost-vs-lightgbm Acesso em: 12 julho. 2022.

Welcome to LightGBM´s documentation (2022). Contents. Disponível em: https://lightgbm.readthedocs.io/en/latest/ Acesso em: 12 julho. 2022. Microsoft Corporation. Revision 17bfe1a1

70% dos Brasileiros pretendem continuar comprando online após a pandemia. SBVC. Sociedade Brasileira de Varejo e Consumo, 24 maio. 2020. Notícias. Disponível em: https://sbvc.com.br/brasileiros-online-apos-pandemia/. Acesso em: 12 julho.. 2022.