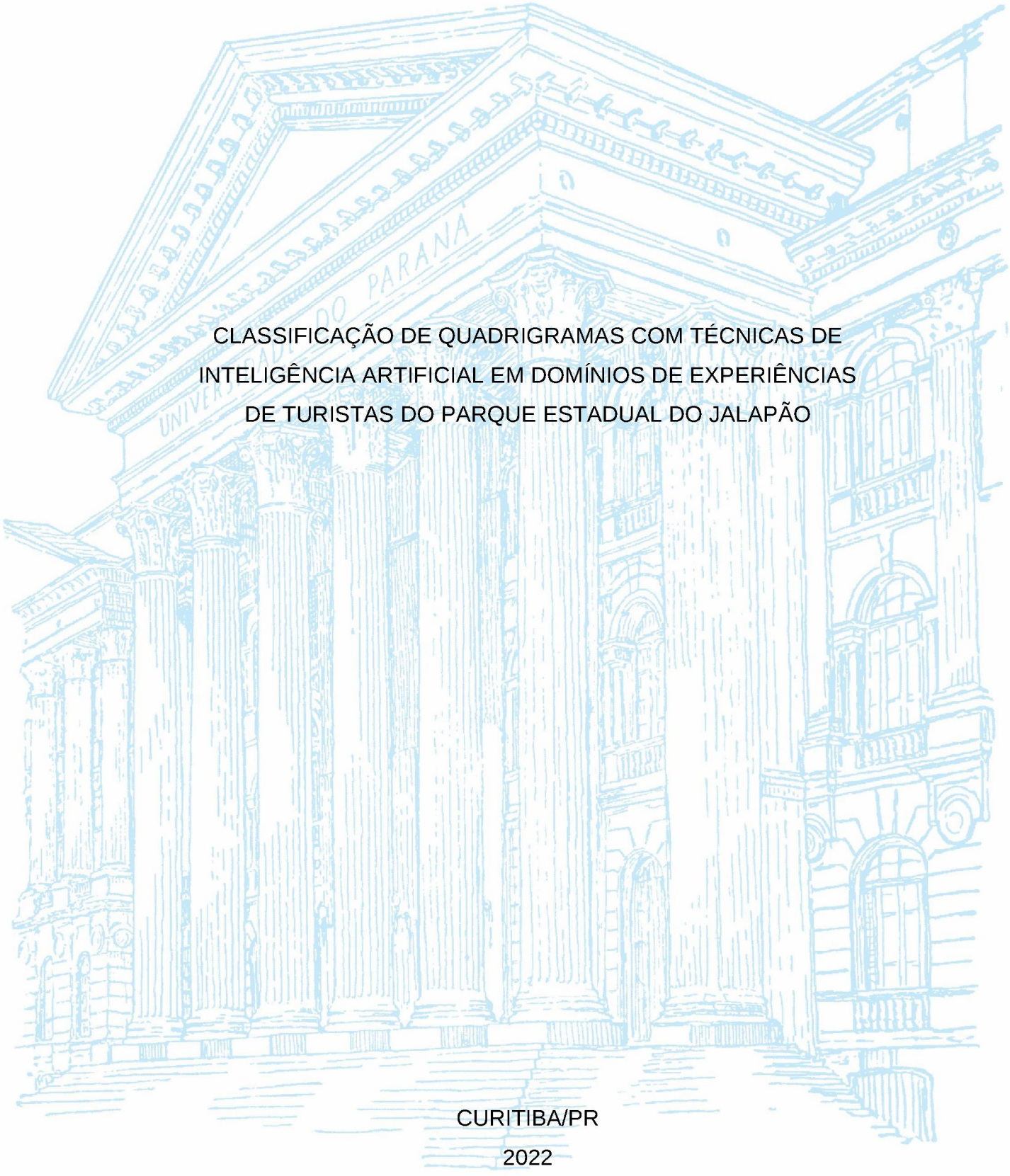
UNIVERSIDADE FEDERAL DO PARANÁ

JHONATHAN DE SOUZA LIMA



CLASSIFICAÇÃO DE QUADRIGRAMAS COM TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL EM DOMÍNIOS DE EXPERIÊNCIAS DE TURISTAS DO PARQUE ESTADUAL DO JALAPÃO

CURITIBA/PR

2022

JHONATHAN DE SOUZA LIMA

CLASSIFICAÇÃO DE QUADRIGRAMAS COM TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL EM DOMÍNIOS DE EXPERIÊNCIAS DE TURISTAS DO PARQUE ESTADUAL DO JALAPÃO

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. João Eugenio Marynowski

CURITIBA/PR

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 **JHONATHAN DE SOUZA LIMA** intitulada: **Classificação De Quadrigramas Com Técnicas De Inteligência Artificial Em Domínios De Experiências De Turistas Do Parque Estadual Do Jalapão**, 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, 22 de Novembro de 2022.

JOÃO EUGENIO MARYNOWSKI
Presidente da Banca Examinadora

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

Classificação de quadrigramas com técnicas de Inteligência Artificial em domínios de experiências de turistas do Parque Estadual do Jalapão

Jhonathan de Souza Lima
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
jhonathansouza93@ufpr.br

João Eugenio Marynowski
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
jeugenio@ufpr.br

***Resumo*—O turismo é uma área fundamental para o bem-estar das pessoas e economia de um munícipio. Sendo assim, é possível promover maior qualidade de vida para os cidadãos e melhorias em uma região com pontos turísticos. Desta forma, se faz necessária uma análise sobre as experiências de turistas. Neste contexto, existem trabalhos que analisam manualmente as experiências turísticas através de questionários ou comentários, o que limita a quantidade de dados a ser analisada. Contudo, este trabalho apresenta um experimento de classificação de experiências de turistas em pontos turísticos do Parque Estadual do Jalapão (PEJ). Essa análise é feita atráves de algoritmos de Inteligência Artificial (IA) desenvolvidos em linguagem de programação *Python*. Foram empregados os métodos para classificação de textos: *Gaussian Naive Bayes*, *Support Vector Machine (SVM)* e *Long Short Term Memory (LSTM)* Bidirecional. Por fim, o modelo SVM obteve o melhor resultado dentre eles.**

***Index Terms*—turismo, experiência, comentários, classificação, Jalapão,**

***Abstract*—Tourism is an essential area for the well-being of people and the economy of a city. So it's possible to promote a better life quality for citizens and tourist programs in a city with tourist spots. To perform that promotion is necessary to analyze tourist experiences. In that context, there are works that analyze tourist experiences manually through questionnaires or comments, what limits the amount of data to be analyzed. However, this research presents a trial to classify experiences of tourists in tourist spots inside Jalapão State Park. That analysis was built in Artificial Intelligence (AI) algorithms developed by Python programming language. Methods for text classification were used: Gaussian Naive Bayes, Support Vector Machine (SVM) and Long Short Term Memory (LSTM) Bidirectional. Ultimately, the SVM model had the best result among them.**

***Index Terms*—tourism, experience, comments, classification, Jalapão**

I. Desenvolvimento

O turismo é um fenômeno social que é caracterizado pelo deslocamento temporário até outro local, gerando com isso, relações econômicas, sociais e culturais [1].

A atividade de turismo é um segmento que expõe o patrimônio natural e cultural com o objetivo de despertar a consciência ambientalista e promover o bem-estar local [2]. Dentro desse contexto, existe o conceito de ecoturismo, que é a prática turística em ambientes naturais, com o objetivo de relacionar-se de forma mais próxima com o meio natural e social [3]. Para essa prática é necessário se deslocar para um lugar que soma elementos paisagísticos e atrativos naturais que conectam o turista ao local [3]. Dentre os pontos turísticos nacionais e internacionais que são proeminentes para a prática de ecoturismo, pode-se citar o Parque Estadual do Jalapão (PEJ) [4], situado ao leste do estado do Tocantins, reunindo ricos elementos naturais para o ecoturismo.

Segundo os autores Pine e Gilmore [5], existem quatros domínios da experiência turística, que podem ser classificados da seguinte forma: entretenimento, educação/aprendizagem, evasão/escapismo e estética/contemplação. O domínio do entretenimento diz respeito a uma experiência recreativa; no domínio educação/aprendizagem encontram-se comentários relacionados a geografia, clima, solo, fauna e flora do local; em relação ao domínio do escapismo, diz-se sobre a experiência na qual o turista se desconecta do habitual e cotidiano para um estado de meditação e relaxamento; e por fim, o domínio de Estética compreende a relação de contemplação e deslumbramento dos turistas com as paisagens vistas.

No trabalho proposto por Kaiser et al. [6] utiliza-se de comentários escritos no endereço eletrônico *TripAdvisor* [7] para se determinar manualmente a experiência de turistas em pontos turísticos do PEJ.

Neste trabalho, propõe-se a utilização de técnicas de IA para classificação de textos (quadrigramas) referentes a esses comentários. Dentre os algoritmos/métodos de classificação mais populares utilizados para classificação de textos, citados por Gurinder et al. [8], são usados neste trabalho: *Support Vector Machine* (SVM), *Gaussian Naive Bayes* e Redes Neurais Recorrentes (RNR) com memória longeva de curto prazo, conhecida como *Long Short-Term Memory* (LSTM).

O objetivo do trabalho é classificar os comentários públicos sobre os pontos turísticos Cachoeira da Velha, Cachoeira da Formiga, Fervedouro, Dunas do Jalapão, Povoado de Mumbuca e Serra do Espírito Santo, dentro dos quatros domínios de experiência apresentados.

A classificação automatizada de texto tem sido considerada vital para gerenciar e processar uma grande quantidade de

documentos em formato digital que são difundidos e estão em constante crescimento. Normalmente, a maioria dos dados para classificação de gêneros são coletados da *web*, através de grupos de notícias, boletins e notícias transmitidas ou impressas [9]. O objetivo da classificação de texto é a categorização de documentos em um número fixo de categorias predeterminadas. Cada documento estará em uma única, nenhuma ou várias categorias. Utilizando aprendizado de máquina, o principal objetivo é aprender classificadores através de instâncias que realizam as atribuições de categoria automaticamente [9]. Muitos estudiosos estudaram a tecnologia de classificação de texto usando dois métodos principais, incluindo o aprendizado de máquina tradicional e o aprendizagem profundo, que é popular atualmente [10].

A. Descrição dos dados

Primeiramente, salienta-se que utilizou-se de uma base de dados advinda do trabalho apresentado por Kaiser et al. [6]. A base de dados utilizada foi extraída pelos autores através de uma técnica de captura de dados da *internet* chamada *Web Scraping*. Os comentários foram agrupados em quadrigramas [6].

Um quadrigrama é um grupo de quatro palavras, que segundo Hyland [11] carrega maior significação e pode ser encontrado com maior frequência em um texto. Tal estrutura de palavras é uma peça fundamental para a aprendizagem de uma linguagem [11].

Na Tab. I são mostrados quadrigramas de acordo com domínios de experiências.

Tabela I
QUADRIGRAMAS CLASSIFICADOS POR DOMÍNIO DE EXPERIÊNCIA

Domínios da experiência	**Exemplo de quadrigrama**	**Ponto turístico**
Educação/Aprendizado	"amante natureza biodiversidade pais"	Cachoeira da Formiga
Educação/Aprendizado	"muita pedra solta areia"	Serra do Espírito Santo
Educação/Aprendizado	"duna formada erosao serra"	Dunas do Jalapão
Entretenimento	"artesanato feito capim dourado"	Povoado de Mumbuca
Entretenimento	"Agua nao deixa afundar"	Fervedouro
Entretenimento	"nao possível entrar agua"	Cachoeira da Velha
Estética/contemplação	"cachoeira maravilhosa agua azul"	Cachoeira da Formiga
Estética/contemplação	"madrugada ver nascer sol"	Serra do Espírito Santo
Estética/contemplação	"serra espirito santo fundo"	Dunas do Jalapão
Evasão/Escapismo	"Nao da vontade sair"	Fervedouro
Evasão/Escapismo	"nao quer ir embora"	Cachoeira da Formiga
Evasão/Escapismo	"agua sentir grandeza natureza"	Cachoeira da Velha

B. Métodos

A maioria dos algoritmos de classificação de texto usa um modelo de pacote de palavras em combinação com modelos probabilísticos de Bayes, redes neurais e técnicas de hiperplano multidimensional para estimar a probabilidade de um texto pertencer a uma classe [12].

A classificação de texto com o auxílio de técnicas de IA pode ser alcançada usando muitos algoritmos de aprendizado de máquina desenvolvidos para essas tarefas ao longo de vários anos [8]. Esses algoritmos provaram atingir altas precisões em previsões e são, portanto, altamente confiáveis [8].

Para realizar a classificação dos domínios de experiência foi necessário aplicar técnicas de vetorização de texto. Como os dados de texto não podem ser usados diretamente para o treinamento de parâmetros de um modelo de classificação, é necessário vetorizar os dados do texto original e torná-los numéricos, e então a operação de extração de características pode ser realizada [13].

Neste trabalho, o primeiro passo para a extração de características de texto é o processo chamado de *tokenização*. A *tokenização*, também conhecida como segmentação de palavras, quebra a sequência de caracteres em um texto, localizando o limite de cada palavra, ou seja, os pontos onde uma palavra termina e outra começa. Para fins de linguística computacional, as palavras assim identificadas são frequentemente chamadas de *tokens* [14]. Ou seja, a *tokenização* é a transformação de texto legível por humanos em *tokens* legíveis por máquina [15].

Dentre as técnicas de vetorização de palavras existe o TF-IDF (*Term Frequency–Inverse Document Frequency*) [13]. O TF-IDF pode ser formulado pela multiplicação entre a frequência do termo e a frequência inversa de documentos. A Frequência do Termo (TF) é a medição de quão frequentemente um termo ocorre em um documento [16]. A Frequência Inversa de Documentos (IDF) mede a importância de um termo dentro de um conjunto de documentos [17].

Nas Eq. (1) e Eq. (2) são mostradas a frequência do termo e a frequência inversa de documentos, respectivamente.

$$TF = \frac{FPD}{NPD} \tag{1}$$

Onde FPD significa a frequência do termo no documento e NPD é o número de termos no documento.

$$IDF = \log_e(1 + \frac{ND}{NDP}) \tag{2}$$

Onde ND significa o número de documentos e NDP é o número de documentos com o termo.

O TF-IDF é expresso como um conjunto de pontuações. Essas pontuações são calculadas multiplicando TF e IDF para termos específicos. Então, a pontuação de qualquer termo em qualquer documento pode ser representada pela Eq. (3) [18].

$$TFIDF = TF \times IDF \tag{3}$$

Onde TF é a frequência do termo e IDF é a frequência Inversa de Documentos.

Na Tab. II são mostrados valores TF-IDF de alguns quadrigramas extraídos da base de dados.

Tabela II
VALORES DE TF-IDF PARA CADA DOMÍNIO DE EXPERIÊNCIA E QUADRIGRAMA ESPECÍFICO

Domínio de experiência	Quadrigrama	Valor TF-IDF
Educação/Aprendizado	sorvete fruta cerrado tem	0.516395958135
Entretenimento	visita trilha suspensa facil	0.492845919043
Evasão/Escapismo	jalapao bruto todo repetem	0.476863555460
Estética/Contemplação	linda beleza lembra mini	0.461324794431

Após o processo de *tokenização*, também foi usado um conceito chamado *Word Embedding* para emprego do método Bi-LSTM. Os *Word Embedding* são modelos matemáticos que codificam relações de palavras dentro de um espaço vetorial através do processo de treinamento baseado em informações de coocorrência entre palavras [19].

A sequência de etapas usada neste trabalho para classificação de quadrigramas é ilustrada na Fig. 1.

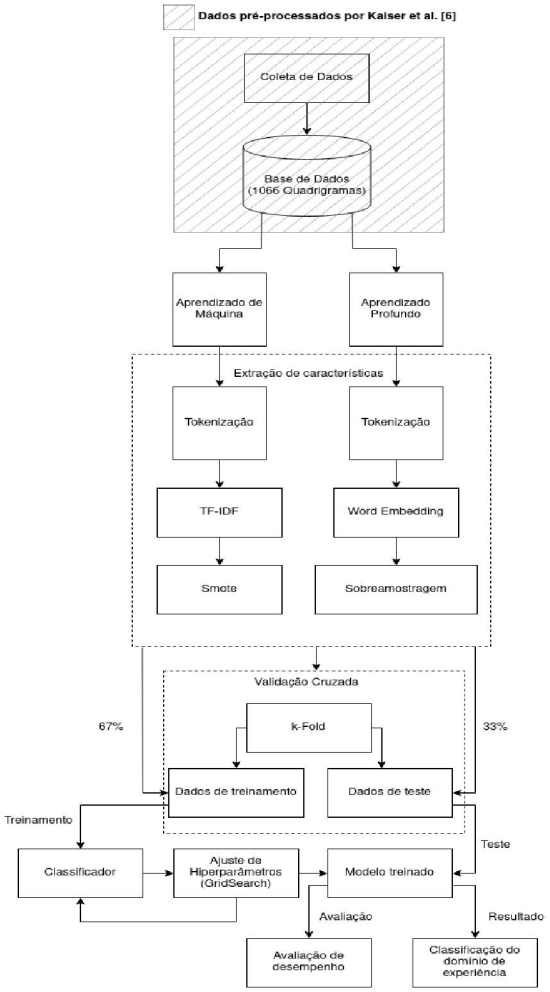


Figura 1. Etapas para classificação de quadrigramas através de IA.

Este trabalho está dividido em cinco etapas, que são: 1) Coleta de dados; 2) Preparação do treinamento; 3) Treinamento; 4) Ajuste de Hiperparâmetros e 5) Predição. Os algoritmos desenvolvidos apresentaram modelos de predição para classificação de quadigramas de comentários de turistas no domínio de experiência correspondente.

1) *Coleta de dados*: A coleta de dados foi feita a partir de uma base de dados oriunda do trabalho realizado pelos autores Kaiser et al. [6]. O pré-processamento necessário em classificação de textos foi realizado por Kaiser et al. [6]. Os comentários dos turistas foram extraídos da plataforma digital *TripAdvisor* [7], sem expor as identidades pessoais [6].
Os quadrigramas que ocorreram no mínimo 2 vezes foram reunidos através dos comentários dos pontos turísticos Cachoeira da Velha, Cachoeira da Formiga, Fervedouro, Dunas do Jalapão, Povoado de Mumbuca e Serra do Espírito Santo, de acordo com as quantidades da Tab. III.

Tabela III
QUANTIDADE DE QUADRIGRAMAS POR PONTO TURÍSTICO

Ponto turístico	Número de quadrigramas
Povoado de Mumbuca	172
Cachoeira da Velha	173
Dunas do Jalapão	184
Serra do Espírito Santo	183
Fervedouro	179
Cachoeira da Formiga	175

A base utilizada aqui então totaliza 1.066 registros de quadrigramas referentes aos comentários sobre o PEJ, na seguinte ordem: Educação/Aprendizagem (282 quadrigramas), Entretenimento (451 quadrigramas), Estética/Contemplação (274 quadrigramas) e Evasão/Escapismo (59 quadrigramas) [6], como mostrado na Tab. IV. Salienta-se que os quadrigramas foram classificados manualmente pelos autores Kaiser et al. [6].

Tabela IV
QUANTIDADE DE QUADRIGRAMAS POR DOMÍNIO DE EXPERIÊNCIA

Domínios da experiência	Quadrigramas	Porcentagem
Educação/Aprendizado	282	26,454%
Entretenimento	451	42,307%
Estética/contemplação	274	25,704%
Evasão	59	5,535%
TOTAL	**1.066**	**100%**

2) *Preparação do treinamento*: A preparação para o treinamento é a etapa que consiste em escolher a melhor divisão entre os dados de treinamento e teste/validação do modelo de classificação. Como a base de dados é pequena a melhor escolha é usar a técnica de validação cruzada [20].
Neste trabalho, é usada a técnica de validação cruzada *k-fold*. Na técnica de validação cruzada *k-fold*, os dados são divididos em k partições iguais. O treinamento do modelo é feito em k partes, exceptuando-se uma, e essa parte é usada para teste [20]. A escolha ideal para

treinamento de um modelo é a validação cruzada *k-fold* com grande valor de *k*, porém menor que o número de instâncias [20].
Como mostrado na Fig. 2, itera-se sucessivamente dentro do número de partições predeterminado (k) do conjunto de teste sobre toda base de dados.

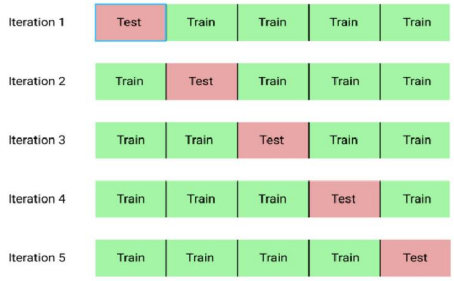


Figura 2. Ilustração de técnica de validação cruzada k-fold [21].

Neste trabalho também foi necessário o uso de uma técnica de sobreamostragem de dados chamada *SMOTE (Synthetic Minority Oversampling Technique)*. Segundo Nitesh V. Chawla [22], o SMOTE é uma popular técnica de balanceamento de dados que gera dados sinteticamente para a classe minoritária balancear os dados de treinamento. Como mostrado na Fig. 3, para cada amostra de classe minoritária x_i (triângulo laranja na Fig. 3), o SMOTE encontra os *k*-vizinhos mais próximos entre as outras amostras de classe minoritária, escolhendo aleatoriamente um $\widehat{x_i}$ deles, e sua distância de x_i é multiplicada por um número aleatório $\delta \in [0, 1]$. A nova amostra resultante x_n(círculo preto na Fig. 3) está localizada entre x_i e o vizinho selecionado $\widehat{x_i}$ [23].

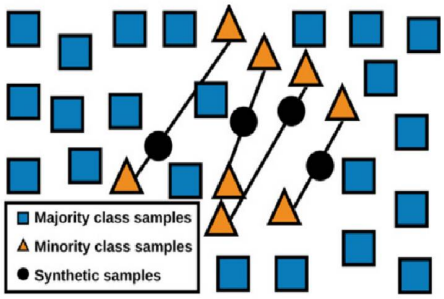


Figura 3. Ilustração da utilização da técnica SMOTE [23].

3) *Treinamento*: Os modelos de classificação de textos foram utilizados com uma biblioteca do *Python* chamada *Scikit-Learn*; ou um *framework* de IA (conjunto de códigos genéricos) chamado *TensorFlow*, juntamente com a API (*Application Programming Interface*) *Keras*. Nesta API existem funções previamente construídas para o treinamento e teste de um modelo de classificação.
Os modelos de classificação utilizados no trabalho foram o SVM, *Gaussian Naive Bayes* e Bi-LSTM, que são descritos a seguir.

a) *Support Vector Machine (SVM)*
O modelo SVM é um dos algoritmos de Aprendizado de Máquina mais utilizados para classificação de pequenos conjuntos de dados [11]. O objetivo do modelo é construir um hiperplano entre as amostras utilizadas para o treinamento da rede [11]. Além disso, o modelo visa maximizar a margem entre os vetores de suporte; esses dados são relevantes para descobrir o melhor separador, o hiperplano, encontrando o plano ótimo entre as amostras de diferentes classes [11].
Para um problema com múltiplas classes usando SVM geralmente é usada uma técnica chamada *One-Against-One*, no qual são construídos hiperplanos, onde cada hiperplano é construído pelas amostras de treinamento de duas classes escolhidas dentre múltiplas classes [24].
Na Fig. 4 é mostrado um exemplo de divisão de múltiplas classes por hiperplanos. Em ambos eixos do gráfico encontram-se características distintas e cada ponto (valor de TF-IDF) é segregado dentro de uma classe.

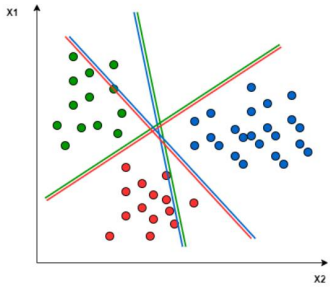


Figura 4. Exemplo de divisão de múltiplas classes por hiperplanos [25].

b) *Gaussian Naive Bayes*
O modelo *Gaussian Naive Bayes* considera os pontos de dados com *n* características em que as características podem ser quaisquer dados de texto. Neste contexto de classificação de texto, a probabilidade de um texto dado Y pertencer a uma classe X é calculado usando o Teorema de *Bayes* [26], de acordo com a Eq. (4).

$$P(X \mid Y) = P(X \mid Y) \times P(X)/P(Y) \qquad (4)$$

Onde P(X | Y) denota a probabilidade de um evento X dado que um evento Y ocorreu; P(Y | X) denota a probabilidade de um evento Y dado que um evento X ocorreu, P(X) é a probabilidade de um evento X acontecer e P(Y) é a probabilidade de um evento Y acontece. Um condicional importante é que P(Y) nunca pode ser igual a zero [26].
A partir do Teorema de *Bayes* , aplicando-se a Regra da Cadeia para *n* eventos, tem-se a probabilidade conjunta para *n* eventos [27].

Em *n*-gramas (neste trabalho usa-se quadrigramas) tem-se uma sequência de palavras vista como $w_1, w_2, ..., w_n$, na qual pode ser formulada como o produto de uma série de probabilidades condicionais, de acordo com a Eq. (5) [27].

$$p(w_1 w_2 ... w_n) = p(w_1) \times P(w_1 \mid w_2) \times \prod p(w_n \mid w_{n-2} w_{n-1}) \quad (5)$$

Uma abordagem comum para calcular as probabilidades *P(X | Y)* é assumir que para cada valor discreto possível C_k de y, a distribuição de cada x_i contínuo é Gaussiano, e é definido por uma média μ_{ki} e desvio padrão σ_{ki} específico para x_i e C_k, de acordo com a Eq. (6) [28]:

$$P(x_i \mid y = C_k) = \frac{1}{\sigma_{ki}\sqrt{2\pi}} \exp\left\{-\frac{(x_i - \mu_{ki})^2}{\sigma_{ki}^2}\right\} \quad (6)$$

c) *Long Short-Term Memory (LSTM)*
O modelo LSTM é um tipo especial de Redes Neurais Recorrentes. As Redes Neurais Recorrentes têm ciclos de realimentação em sua arquitetura, permitindo que as informações sejam memorizadas por curtos termos [29]. A saída de uma RNR em um estágio depende da saída anterior e da entrada atual [29].
As Redes Neurais Recorrentes têm memórias de curto prazo, então sua computação pode ser lenta, além dessas memórias sofrerem com o problema do gradiente de fuga [29].
Entretanto, os autores Hochreiter e Schmidhuber [30] desenvolveram um modelo longevo de memória de curto prazo para resolver os problemas com a estrutura RNR. Para superar esses problemas, redes neurais adicionais chamadas *gates* são introduzidas, que lidam com o fluxo de informações na rede [29].
Na Fig. 5 é mostrada a estrutura de um modelo de rede neural LSTM, onde o bloco com denominação X_t é o valor de base; H_{t-1} é o estado da camada escondida anterior; H_t é o estado da camada escondida atual; C_{t-1} é o estado da célula anterior; C_t é o estado da célula atual (dado de saída do modelo); os blocos em cor amarelha são funções de ativação Tangente Hiperbólica; os blocos na cor vermelha são funcões de ativação Sigmoide; e os blocos de conexão, cuja cor é rosa, realizam operações matemáticas de adição ou multiplicação de valores [31].

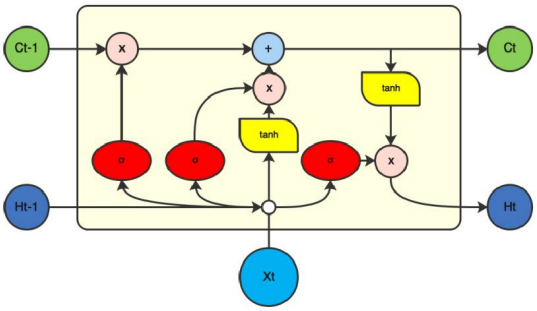


Figura 5. Estrutura de um modelo LSTM [32].

Entretanto, neste trabalho fez-se necessário o uso do modelo de rede neural LSTM Bidirecional. A rede neural Bi-LSTM é composta por unidades LSTM que operam em ambas as direções para incorporar informações de contexto passadas e futuras. O Bi-LSTM pode aprender problemas de modelagem sequencial de dependências de longo prazo e é amplamente usado para classificação de texto. Ao contrário da rede LSTM, a rede Bi-LSTM possui duas camadas paralelas que se propagam em duas direções com passagens diretas e reversas para capturar dependências em dois contextos [33]. A estrutura da rede Bi-LSTM é mostrada na Fig. 6

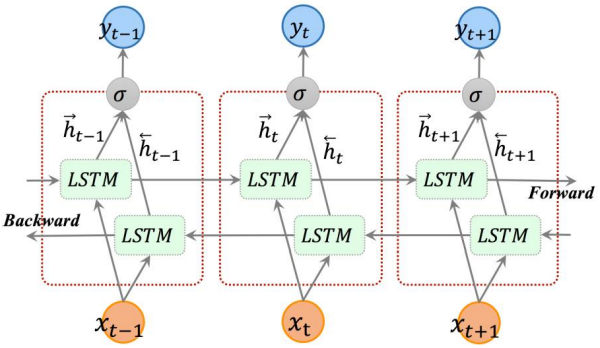


Figura 6. Estrutura de um modelo LSTM Bidirecional [34].

4) Ajuste de Hiperparâmetros: Em geral, construir um modelo de aprendizado de máquina ou aprendizado profundo eficaz é um processo complexo e demorado que envolve a determinação de um algoritmo apropriado e a obtenção de uma arquitetura de modelo ideal por ajuste de seus hiperparâmetros [35].
Neste trabalho é usado uma técnica para ajuste de hiperparâmetros, chamada *Grid Search* ou Pesquisa em Grade que é um dos métodos mais usados para explorar o espaço de configuração de hiperparâmetros. A pesquisa em grade pode ser considerada uma busca ativa ou um método de força bruta que avalia todas as combinações de hiperparâmetros dadas a grade de configurações. A pesquisa em grade trabalha avaliando o produto cartesiano de um conjunto finito de valores especificados pelo

usuário [35]. Os valores utilizados e resultados obtidos são apresentados na Seção Resultados.

5) *Predição*: São usadas funções das bibliotecas do *Python* para avaliação da precisão do modelo, nas quais existem métricas de avaliação como a matriz de confusão, valores de precisão, *recall* e *F1-score* do modelo de classificação em dados de teste e validação.
Como a predição é do tipo categórica com múltiplas classes, a matriz de confusão multiclasse é usada para avaliação de cada modelo, como mostrado na Tab. V.

Tabela V
MATRIX DE CONFUSÃO PARA MÚLTIPLAS CLASSES

		Classe predita			
		Classe 1	**Classe 2**	**...**	**Classe n**
Classe Atual	**Classe 1**	X_{11}	X_{12}	...	X_{1n}
	Classe 2	X_{21}	X_{22}	...	X_{2n}
	.	.	.	.	.
	.	.	.	.	.
	.	.	.	.	.
	Classe n	X_{n1}	X_{n2}	...	X_{nn}

Exemplo de matrix de confusão multiclasse [36].

O número total de falsos negativos (TFN) e falsos positivos (TFP) para cada classe i ou j é calculado baseado nas Eq. (7) e Eq. (8) [36]. O número total de verdadeiros positivos é obtido pela Eq. (9) [36]. Onde i e j representam índices de classes distintas.

$$TFN_i = \sum_{j=1, j\neq i}^{n} X_{ij} \tag{7}$$

$$TFP_i = \sum_{j=1, j\neq i}^{n} X_{ji} \tag{8}$$

$$TTP_{all} = \sum_{j=1}^{n} X_{jj} \tag{9}$$

Para computar a Precisão (P) e *Recall* (R) para cada classe i ou j [36], tem-se as Eq. (10) e Eq. (11).

$$P_i = \frac{TTP_{all}}{TTP_{all} + TFP_i} \tag{10}$$

$$R_i = \frac{TTP_{all}}{TTP_{all} + TFN_i} \tag{11}$$

Outra métrica a ser utilizada é o *F1-score*, que é a média harmônica de precisão e *recall*, na qual pode ser expressa pela Eq. (12) [37].

$$F1\text{-}score_i = 2 \times \frac{P_i \times R_i}{P_i + R_i} \tag{12}$$

C. Tecnologias

Os algoritmos foram desenvolvidos em um computador pessoal com processador Ryzen 9 5900 com 12 núcleos e 24 *threads* de processamento em uma frequência máxima de 4,8 GHz em cada núcleo, Placa de vídeo RTX 3080 com 10 GB de VRAM GDDR6X e GPU (*Graphical Processing Units*) com 8.704 *cuda cores* e 272 *tensor cores* de processamento em uma frequência máxima de 1,71 GHz em cada núcleo e sistema operacional Ubuntu 20.04.4 LTS. Foram utilizadas as seguintes aplicações e bibliotecas:

- Linguagem de programação *Python*, versao 3.9;
- Biblioteca *scikit-learn*, versão 1.1.2;
- *Framework* de IA *TensorFlow-GPU*, versão 2.0;
- API *Keras*, versão 2.9.0.

O uso de GPUs tem transformado a maneira na qual redes neurais têm sido treinadas com técnicas de *machine learning* e *deep learning*, atingindo entre 10 a 100 vezes mais desempenho se comparadas com sistemas de computação convencionais, devido a computação paralela e a capacidade em realizar múltiplas operações matemáticas matriciais ao mesmo tempo, pois incluem centenas/milhares de núcleos ou *cores* na arquitetura de *hardware*, enquanto que CPUs (*Central Processing Units*) são otimizadas para baixas latências e tem melhor desempenho em processamentos sequencias do que GPUs [38].

Neste trabalho, de acordo com as tecnologias de alto desempenho apresentadas, foram usadas as quantidades de 12 núcleos e 24 *threads* de processamento da CPU, assim como 8.704 *cuda cores* e 272 *tensor cores* de processamento da GPU, com máxima frequência em ambos os *hardwares*. Portanto, a configuração de *hardware* usada para desenvolver as técnicas de inteligência artificial apresentadas neste trabalho, possibilitaram a realização das tarefas em menos tempo do que em computadores convencionais.

O *Keras* é uma biblioteca de rede neural (*open-source*) e codificação em linguagem de programação *Python* que pode ser executado na maioria das APIs alto nível de *TensorFlow*, *Theano* ou *Microsoft Cognitive Toolkit*. O Keras foi desenvolvido para facilitar a implementação usando redes neurais de aprendizado profundo com uma abordagem orientada ao usuário, modular e extensível [39].

A biblioteca *Keras* compreende várias operações baseadas em redes neurais, incluindo tipos de camadas, funções de ativação, otimizadores e ferramentas de hospedagem que a tornam adequeda para trabalhos com imagens e mineração de texto. O *Keras* suporta convolução, redes neurais persistentes e camadas de utilidades gerais suplementares (*dropout*, *batch normalization*, e *polling*) [39].

II. RESULTADOS E DISCUSSÕES

A. Métricas para avaliação dos desempenhos dos modelos de classificação com dados desbalanceados

Observou-se que a base de dados utilizada contém dados desbalanceados como mostrado na Fig. 7.

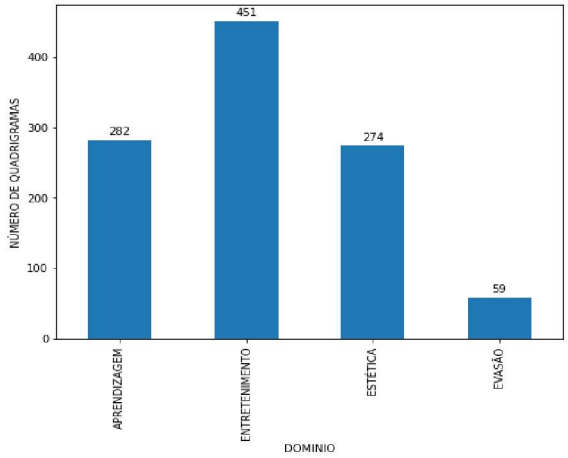


Figura 7. Quantidade de quadrigramas com base de dados desbalanceada (base de dados original).

Na Fig. 7 observa-se que nos dados originais existem 282 quadrigramas no domínio de Aprendizagem, 451 quadrigramas no domínio de Entretenimento, 274 quadrigramas no domínio de Estética e 59 quadrigramas no domínio de Evasão.

De acordo com D. Makienko et al. [40], mostra-se que a qualidade dos modelos preditivos treinados em dados desequilibrados podem depender do grau de desequilíbrio e para algumas amostras o desequilíbrio pode afetar drasticamente a qualidade da classificação.

Segundo S. Kotsiantis et al. [41], a relação entre o tamanho do conjunto de treinamento e o desempenho de classificação inadequado para conjuntos de dados desbalanceados pode ser que em pequenos conjuntos de dados desbalanceados a classe minoritária é mal representada por um número excessivamente reduzido de exemplos que podem não ser suficiente para a aprendizagem.

Tendo ciência que a classificação com dados desbalanceados pode ser drasticamente danosa, neste trabalho utiliza-se dos resultados da classificação com dados desbalanceados meramente para fins de comparação com resultados oriundos de técnicas de tratamento de dados, observando-se as melhorias dessas técnicas para cada modelo de classificação.

Originalmente, distribuiu-se a base de dados entre treinamento (67%) e teste (33%) em cada modelo de classificação. Porém, observou-se a necessidade de distribuir os dados de teste usando a técnica de validação cruzada.

Para fins de comparação, os resultados dos modelos de classificação de texto estudados neste trabalho são apresentados em forma de matriz de confusão com múltiplas classes.

Na matriz de confusão são apresentados os domínios da experiência divididos em quadrigramas da base de dados e os que são preditos pelos modelos de classificação. De acordo com o plano cartesiano, no eixo das ordenadas são apresentados os quadrigramas da base de dados em seus respectivos domínios de experiência, e no eixo das abscissas são mostrados os quadrigramas classificados de acordo com a predição do modelo de classificação.

Portanto, o resultado ideal é que os registros de quadrigramas dentro de uma matriz de confusão se concentrem na diagonal principal, e que os registros de quadrigramas enquadrados dentro de um domínio de experiência no eixo das ordenadas estejam também no eixo das abscissas.

O mapa de calor na matriz de confusão indica quão preciso o modelo de classificação se apresenta. Quanto mais clara a tonalidade da cor do quadrante de um domínio de experiência, mais quadrigramas foram preditos dentro do domínio correto, e consequentemente, quanto mais escuro, mais quadrigramas foram preditos dentro do domínio incorreto.

Usando o modelo de classificação *Gaussian Naive Bayes* com a base de dados desbalanceada tem-se a matriz de confusão apresentada na Fig.8.

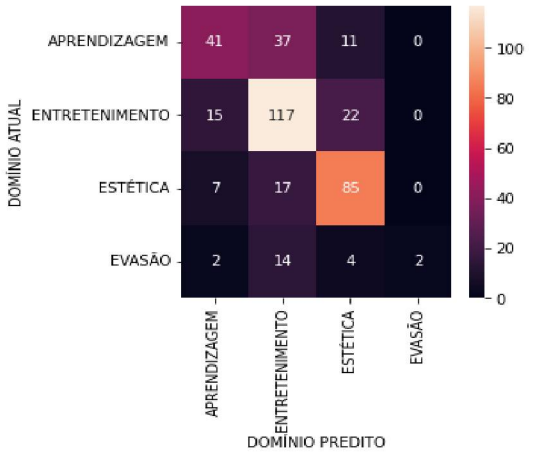


Figura 8. Matriz de confusão com base de dados desbalanceada de treinamento para modelo *Gaussian Naive Bayes*.

Na Fig.8 observa-se que em 65 quadrigramas de Aprendizagem, 41 foram classificados corretamente neste domínio, 15 classificados erroneamente no domínio do Entretenimento, 7 no domínio de Estética e 2 no domínio de Evasão. Em 185 quadrigramas de Entretenimento, 117 foram classificados corretamente neste domínio, 37 classificados erroneamente no domínio de Aprendizagem, 17 no domínio de Estética e 14 no domínio de Evasão. Em 122 quadrigramas de Estética, 85 foram classificados corretamente neste domínio, 11 classificados erroneamente no domínio de Aprendizagem, 22 no domínio de Entretenimento e 4 no domínio de Evasão. Em 2 quadrigramas de Evasão, 2 foram classificados corretamente neste domínio.

Na Tab.VI são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo *Gaussian Naive Bayes*, bem como cada domínio de experiência. Nesta tabela são mostrados os valores de precisão, *recall* e *F1-score* para cada domínio de experiência.

Tabela VI
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO GAUSSIAN NAIVE BAYES COM DADOS DESBALANCEADOS

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	63%	46%	53%
Entretenimento	63%	76%	69%
Estética/Contemplação	70%	78%	74%
Evasão/Escapismo	100%	9%	17%
MÉDIA	**74%**	**52%**	**53%**

Observa-se que a precisão do modelo foi 74%, *recall* de 52% e *F1-score* de 53%.

Usando o modelo de classificação SVM com a base de dados desbalanceada tem-se a matriz de confusão apresentada na Fig.9.

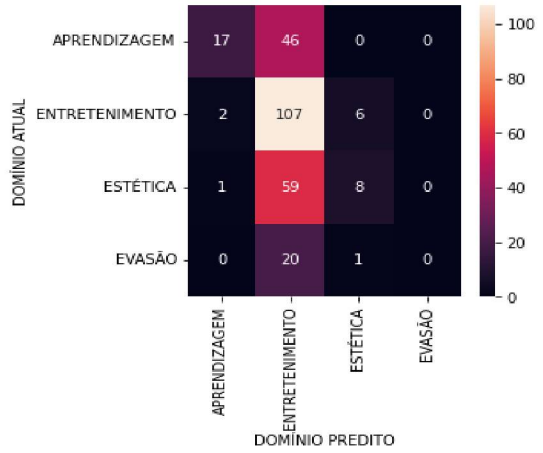


Figura 9. Matriz de confusão com base de dados desbalanceada de treinamento para modelo SVM.

Na Fig.9 observa-se que em 20 quadrigramas de Aprendizagem, 17 foram classificados corretamente neste domínio, 2 classificados erroneamente no domínio do Entretenimento e 1 no domínio de Estética. Em 232 quadrigramas de Entretenimento, 107 foram classificados corretamente neste domínio, 46 classificados erroneamente no domínio de Aprendizagem, 59 no domínio de Estética e 20 no domínio de Evasão. Em 15 quadrigramas de Estética, 8 foram classificados corretamente neste domínio, 6 classificados erroneamente no domínio do Entretenimento e 1 no domínio de Evasão. Nenhum quadrigrama no domínio de Evasão foi selecionado.

Na Tab.VII são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo SVM, bem como cada domínio de experiência. Nesta tabela são mostrados os valores de precisão, *recall* e *F1-Score* para cada domínio de experiência.

Tabela VII
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO SVM COM DADOS DESBALANCEADOS

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	85%	27%	41%
Entretenimento	46%	93%	62%
Estética/Contemplação	53%	12%	19%
Evasão/Escapismo	0%	0%	0%
MÉDIA	**46%**	**33%**	**30%**

Observa-se que a precisão do modelo foi 46%, *recall* de 33% e *F1-score* de 30%.

A rede neural Bi-LSTM foi construída com uma camada de *Embedding* que tem como entrada um vocabulário com 50.000 palavras e comprimento de 1.066 unidades; uma camada de *Spatial Dropout 1D* com 20%; uma camada LSTM bidirecional com 8 unidades (camadas ocultas), na qual tem uma função de ativação *Relu* com 20% de *Dropout* e 20% de *Recurrent Dropout*; uma camada densa com 4 unidades e função de ativação *Softmax*; por fim a rede neural usa o otimizador de modelo do tipo *Adam*. A Fig.10 ilustra como a arquitetura da rede neural Bi-LSTM foi construída.

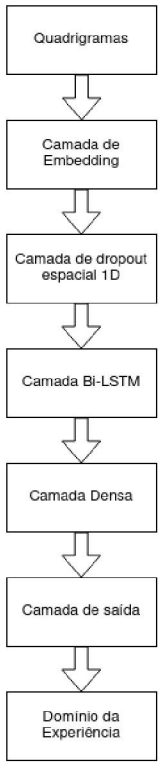


Figura 10. Arquitetura da rede neural Bi-LSTM construída neste trabalho.

Usando o modelo de classificação Bi-LSTM com a base de dados desbalanceada tem-se a matriz de confusão apresentada na Fig.11.

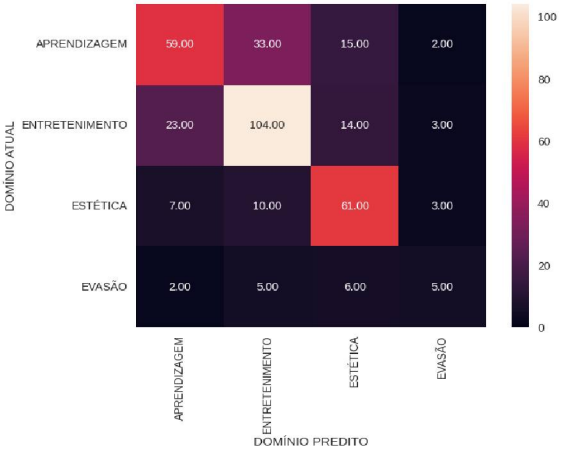


Figura 11. Matriz de confusão com base de bados desbalanceada de treinamento para modelo Bi-LSTM.

Na Fig.11 observa-se que em 91 quadrigramas de Aprendizagem, 59 foram classificados corretamente neste domínio, 23 classificados erroneamente no domínio do Entretenimento, 7 no domínio de Estética e 2 no domínio de Evasão. Em 152 quadrigramas de Entretenimento, 104 foram classificados corretamente neste domínio, 33 classificados erroneamente no domínio de Aprendizagem, 10 no domínio de Estética e 5 no domínio de Evasão. Em 96 quadrigramas de Estética, 61 foram classificados corretamente neste domínio, 15 classificados erroneamente no domínio de Aprendizagem, 14 no domínio de Entretenimento e 6 no domínio de Evasão. Em 13 quadrigramas de Evasão, 5 foram classificados corretamente neste domínio, 2 classificados erroneamente no domínio de Aprendizagem, 3 no domínio de Entretenimento e 3 no domínio de Estética.

Na Tab.VIII são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo Bi-LSTM, bem como cada domínio de experiência. Nesta tabela são mostrados os valores de precisão, *recall* e *F1-Score* para cada domínio de experiência.

Tabela VIII
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO BI-LSTM COM DADOS DESBALANCEADOS

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	54%	65%	59%
Entretenimento	72%	68%	70%
Estética/Contemplação	75%	64%	69%
Evasão/Escapismo	28%	38%	32%
MÉDIA	**57%**	**59%**	**58%**

Observa-se que a precisão do modelo foi 57%, *recall* de 59% e *F1-score* de 58%.

B. Métricas para avaliação dos desempenhos dos modelos de classificação com dados desbalanceados e validação cruzada k-fold

Nesta seção foi usada a técnica de validação cruzada com o número de 10 partições (k = 10) na base de dados. Na Fig.12 é mostrado o resultado usando essa técnica para o modelo *Gaussian Naive Bayes*.

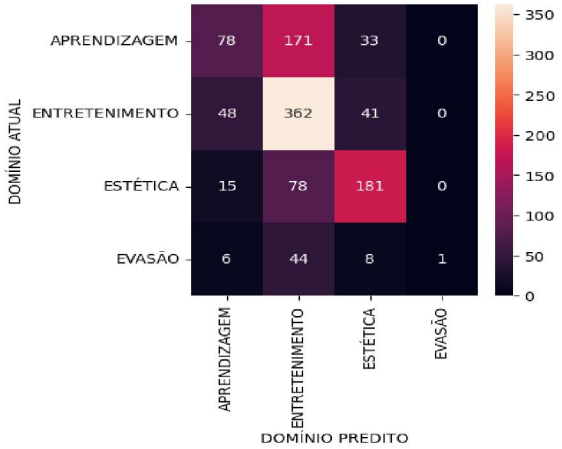


Figura 12. Matriz de Confusão com base de dados balanceada usando validação cruzada *k-fold* (k = 10) para modelo *Gaussian Naive Bayes*.

Na Fig.12 observa-se que em 147 quadrigramas de Aprendizagem, 78 foram classificados corretamente neste domínio, 48 classificados erroneamente no domínio do Entretenimento, 15 no domínio de Estética e 6 no domínio de Evasão. Em 655 quadrigramas de Entretenimento, 362 foram classificados corretamente neste domínio, 171 classificados erroneamente no domínio de Aprendizagem, 78 no domínio de Estética e 44 no domínio de Evasão. Em 263 quadrigramas de Estética, 181 foram classificados corretamente neste domínio, 33 classificados erroneamente no domínio de Aprendizagem, 41 no domínio de Entretenimento e 8 no domínio de Evasão. Em 1 quadrigrama de Evasão, 1 foi classificado corretamente neste domínio.

Na Tab.IX são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo *Gaussian Naive Bayes*, bem como cada domínio de experiência, com validação cruzada *K-Fold* (k = 10) .

Tabela IX
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO *GAUSSIAN NAIVE BAYES* COM VALIDAÇÃO CRUZADA *K-FOLD* (K = 10)

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	53%	28%	36%
Entretenimento	55%	80%	65%
Estética/Contemplação	69%	66%	67%
Evasão/Escapismo	100%	2%	3%
MÉDIA	**69%**	**58%**	**55%**

Observa-se que a precisão do modelo foi 69%, *recall* de 58% e *F1-score* de 55%.

Na Fig.13 é mostrado o resultado usando a técnica de validação cruzada com o número de 10 partições (k = 10) na base de dados para o modelo SVM.

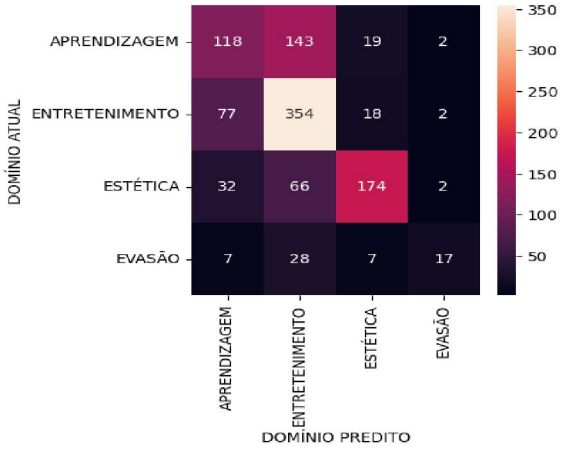


Figura 13. Matriz de Confusão com base de dados balanceada usando validação cruzada *k-fold* (k = 10) para modelo SVM.

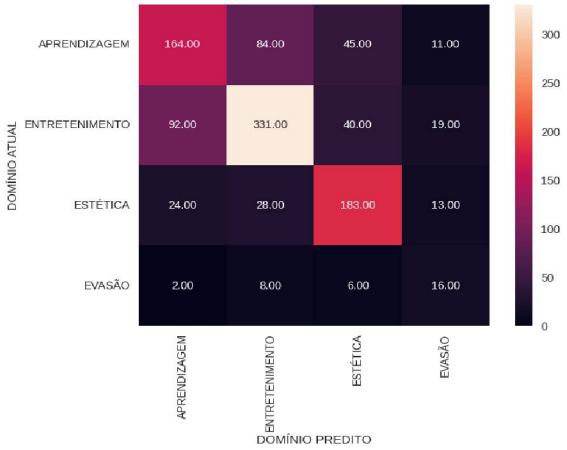


Figura 14. Matriz de Confusão com base de dados balanceada usando validação cruzada *k-fold* (k = 10) para modelo Bi-LSTM.

Na Fig.13 observa-se que em 234 quadrigramas de Aprendizagem, 118 foram classificados corretamente neste domínio, 77 classificados erroneamente no domínio do Entretenimento, 32 no domínio de Estética e 7 no domínio de Evasão. Em 591 quadrigramas de Entretenimento, 354 foram classificados corretamente neste domínio, 143 classificados erroneamente no domínio de Aprendizagem, 66 no domínio de Estética e 28 no domínio de Evasão. Em 191 quadrigramas de Estética, 174 foram classificados corretamente neste domínio, 19 classificados erroneamente no domínio de Aprendizagem, 18 no domínio de Entretenimento e 7 no domínio de Evasão. Em 23 quadrigramas de Evasão, 17 foram classificados corretamente neste domínio, 2 classificados erroneamente no domínio de Aprendizagem, 2 no domínio de Entretenimento e 2 no domínio de Estética.

Na Tab.X são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo SVM, bem como cada domínio de experiência, com validação cruzada *k-fold* (k = 10).

Tabela X
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO SVM COM VALIDAÇÃO CRUZADA *K-FOLD* (K = 10)

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	50%	42%	46%
Entretenimento	60%	78%	68%
Estética/Contemplação	80%	64%	71%
Evasão/Escapismo	74%	29%	41%
MÉDIA	**66%**	**53%**	**56%**

Observa-se que a precisão do modelo foi 66%, *recall* de 53% e *F1-score* de 56%.

Na Fig.14 observa-se que em 282 quadrigramas de Aprendizagem, 164 foram classificados corretamente neste domínio, 92 classificados erroneamente no domínio do Entretenimento, 24 no domínio de Estética e 2 no domínio de Evasão. Em 451 quadrigramas de Entretenimento, 331 foram classificados corretamente neste domínio, 84 classificados erroneamente no domínio de Aprendizagem, 28 no domínio de Estética e 8 no domínio de Evasão. Em 274 quadrigramas de Estética, 183 foram classificados corretamente neste domínio, 45 classificados erroneamente no domínio de Aprendizagem, 40 no domínio de Entretenimento e 6 no domínio de Evasão. Em 59 quadrigramas de Evasão, 16 foram classificados corretamente neste domínio, 11 classificados erroneamente no domínio de Aprendizagem, 19 no domínio de Entretenimento e 13 no domínio de Estética.

Na Tab.XI são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo Bi-LSTM, bem como cada domínio de experiência com validação cruzada *k-fold* (k = 10).

Tabela XI
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO BI-LSTM COM VALIDAÇÃO CRUZADA *K-FOLD* (K = 10)

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	54%	58%	56%
Entretenimento	69%	73%	71%
Estética/Contemplação	74%	67%	70%
Evasão/Escapismo	50%	27%	35%
MÉDIA	**62%**	**56%**	**58%**

Observa-se que a precisão do modelo foi 62%, *recall* de 56% e *F1-score* de 58%.

C. Métricas para avaliação dos desempenhos dos modelos de classificação com dados balanceados, validação cruzada k-fold *e ajuste de hiperparâmetros*

Como mostrado na Tab.IV, há uma má distribuição das classes apresentadas e necessita-se que as quantidades de quadrigramas dos domínios de experiência de Aprendizagem,

Estética e Evasão sejam equiparadas com o domínio de Entretenimento. Dessa forma, foi usada a técnica de balanceamento de classes com SMOTE. O resultado da aplicação da técnica SMOTE é apresentado na Fig.15.

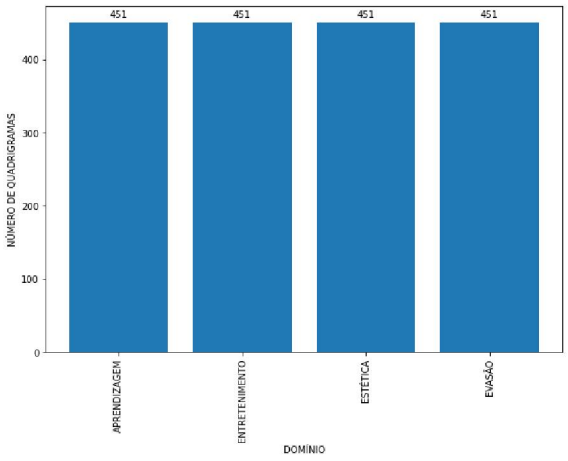


Figura 15. Quantidade de quadrigramas com base de dados balanceada.

Na Fig.15 observa-se que existem em média 451 quadrigramas em cada domínio de experiência depois da aplicação de sobreamostragem.

Aplicando-se a técnica SMOTE para balanceamento de classes no modelo de classificação *Gaussian Naive Bayes* tem-se a matriz de confusão apresentada na Fig.16. Nesta seção foi usada a técnica de validação cruzada com o número de 50 partições (k = 50) na base de dados para aumento do desempenho dos modelos de classificação, pois de um modo geral, quanto maior o valor de k, maior a precisão na validação cruzada [20].

Ajustando-se os hiperparâmetros para cada método de classificação, foram usadas as faixas de valores de hiperparâmetros de acordo com a Tab. XII. Os valores usados têm como referência os trabalhos apresentados pelos autores E. Elgeldawi et al. [42], J. N. v. Rijn et al. [43] e N. Luaran et al. [44]

Tabela XII
TABELA COM FAIXAS DE VALORES DE HIPERPARÂMETROS PARA CADA MÉTODO DE CLASSIFICAÇÃO

Método de classificação	**Parâmetro**	**Min/Atributos**	**Max**	**Passo**
Gaussian Naive Bayes	*var_smothing*	$1x10^{-2}$	$1x10^{-15}$	$x10^{-1}$
SVM	C	0,1	1.000	x10
	Gamma	1	0.0001	÷10
	Kernel	RBF	- -	- -
Bi-LSTM	Camadas Ocultas	4	8	$x1^2$
	Tamanho batch	4	8	$x1^2$
	Épocas	10	100	+45
	Dropout Rate	0	0.4	$x10^1$

No resultado da Fig.16 foram utilizadas as técnicas SMOTE, validação cruzada *k-Fold* (k = 50) e ajuste de hiperparâmetros com o modelo de classificação *Gaussian Naive Bayes*.

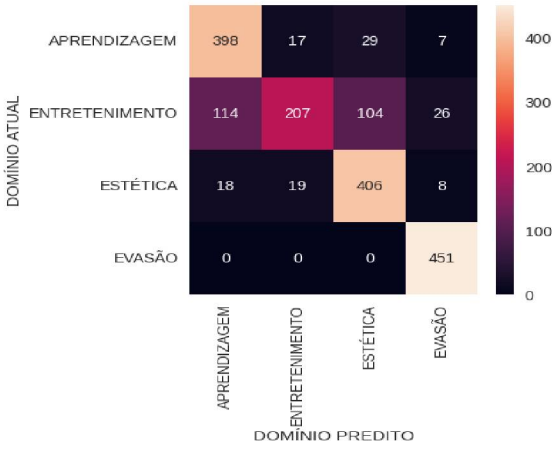


Figura 16. Matriz de Confusão com base de dados balanceada usando SMOTE, validação cruzada *k-fold* (k = 50) e ajuste de hiperparâmetros para modelo *Gaussian Naive Bayes*.

Na Fig.16 observa-se que em 530 quadrigramas de Aprendizagem, 398 foram classificados corretamente neste domínio, 114 classificados erroneamente no domínio do Entretenimento e 18 no domínio de Estética. Em 243 quadrigramas de Entretenimento, 207 foram classificados corretamente neste domínio, 17 classificados erroneamente no domínio de Aprendizagem e 19 no domínio de Estética. Em 539 quadrigramas de Estética, 406 foram classificados corretamente neste domínio, 29 classificados erroneamente no domínio de Aprendizagem e 104 no domínio de Entretenimento. Em 492 quadrigramas de Evasão, 451 foram classificados corretamente neste domínio, 7 classificados erroneamente no domínio de Aprendizagem, 26 no domínio de Entretenimento e 8 no domínio de Estética.

Na Tab.XIII são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo *Gaussian Naive Bayes*, bem como cada domínio de experiência, com SMOTE, validação cruzada *K-Fold* (k = 50) e e ajuste de hiperparâmetros.

Tabela XIII
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO *GAUSSIAN NAIVE BAYES*, VALIDAÇÃO CRUZADA *K-FOLD* (K = 50) E AJUSTE DE HIPERPARÂMETROS

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	75%	88%	81%
Entretenimento	85%	46%	60%
Estética/Contemplação	75%	90%	82%
Evasão/Escapismo	92%	100%	96%
MÉDIA	**82%**	**81%**	**80%**

Observa-se que a precisão do modelo foi 82%, *recall* de 81% e *F1-score* de 80%. Sendo que, de acordo com ajuste de hiperparâmetros obteu-se o valor de 0,0001 para o parâmetro *Smoothing*.

Como mostrado na Fig.17 foram usadas as técnicas SMOTE, validação cruzada *k-fold* (k = 50) e ajuste de hiperparâmetros com o modelo de classificação SVM.

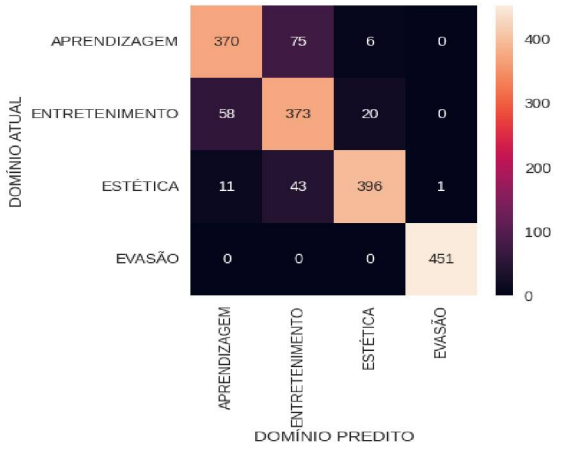


Figura 17. Matriz de Confusão com base de dados balanceada usando SMOTE, validação cruzada *k-fold* (k = 50) e ajuste de hiperparâmetros para modelo SVM.

Na Fig.17 observa-se que em 439 quadrigramas de Aprendizagem, 370 foram classificados corretamente neste domínio 58 classificados erroneamente no domínio do Entretenimento e 11 no domínio de Estética. Em 491 quadrigramas de Entretenimento, 373 foram classificados corretamente neste domínio, 75 classificados erroneamente no domínio de Aprendizagem e 43 no domínio de Estética. Em 422 quadrigramas de Estética, 396 foram classificados corretamente neste domínio, 6 classificados erroneamente no domínio de Aprendizagem e 20 no domínio de Entretenimento. Em 452 quadrigramas de Evasão, 451 foram classificados corretamente neste domínio e 1 no domínio de Estética.

Na Tab.XIV são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo SVM, bem como cada domínio de experiência, com SMOTE, validação cruzada *k-fold* (k = 50) e ajuste de hiperparâmetros.

Tabela XIV
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO SVM COM SMOTE, VALIDAÇÃO CRUZADA *K-FOLD* (K = 50) E AJUSTE DE HIPERPARÂMETROS

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	84%	82%	83%
Entretenimento	76%	83%	79%
Estética/Contemplação	94%	88%	91%
Evasão/Escapismo	100%	100%	100%
MÉDIA	**88%**	**88%**	**88%**

Observa-se que a precisão do modelo foi 88%, *recall* de 88% e *F1-score* de 88%. Sendo que, de acordo com ajuste de hiperparâmetros obteu-se o valor de 10 para o parâmetro Penalidade (C), valor de 0,1 para o parâmetro *Gamma* e argumento *Radial Basis Function (RBF)* para o parâmetro *Kernel*.

Na confecção do modelo de rede neural Bi-LSTM, não foi possível usar a técnica de balanceamento de classes denominada SMOTE, porém foi utilizada uma técnica na qual realizou-se a sobreamostragem dos domínios minotários, que são: domínio da Educação/Aprendizado, Estética/Contemplação e Evasão/Escapismo. Os dados excedentes do domínio majoritário (Entretenimento) foram substituídos de maneira ramdômica por dados de domínios minotários, até que todos os domínios tivessem a mesma quantidade de dados.

Como mostrado na Fig.18 foram usadas as técnicas de sobreamostragem, validação cruzada *k-fold* (k = 50) e ajuste de hiperparâmetros com o modelo de classificação Bi-LSTM.

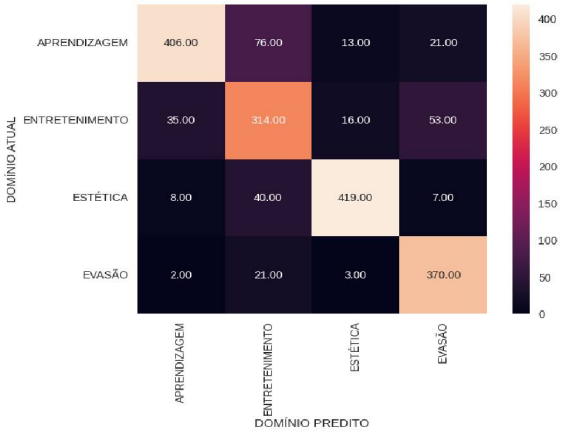


Figura 18. Matriz de Confusão com base de dados balanceada usando as técnicas de sobreamostragem, validação cruzada *k-fold* (k = 50) e ajuste de hiperparâmetros para o modelo Bi-LSTM.

Na Fig.18 observa-se que em 451 quadrigramas de Aprendizagem, 406 foram classificados corretamente neste domínio, 35 classificados erroneamente no domínio do Entretenimento, 8 no domínio de Estética e 2 no domínio de Evasão. Em 451 quadrigramas de Entretenimento, 314 foram classificados corretamente neste domínio, 76 classificados erroneamente no domínio de Aprendizagem, 40 no domínio de Estética e 21 no domínio de Evasão. Em 451 quadrigramas de Estética, 419 foram classificados corretamente neste domínio, 13 classificados erroneamente no domínio de Aprendizagem, 16 no domínio de Entretenimento e 3 no domínio de Evasão. Em 451 quadrigramas de Evasão, 370 foram classificados corretamente neste domínio, 21 classificados erroneamente no domínio de Aprendizagem, 53 no domínio de Entretenimento e 7 no domínio de Estética.

Na Tab.XV são mostrados os resultados de acordo com as métricas para avaliação do desempenho do modelo Bi-LSTM, bem como cada domínio de experiência, com validação cruzada *k-fold* (k = 50).

Observa-se que a precisão do modelo foi 84%, *recall* de 84% e *F1-score* de 84%. Sendo que, de acordo com ajuste de hiperparâmetros obteu-se o valor de 20 épocas, 20% de

Tabela XV
MÉTRICAS PARA AVALIAÇÃO DE DESEMPENHO DO MODELO DE CLASSIFICAÇÃO BI-LSTM COM SMOTE, VALIDAÇÃO CRUZADA *K-FOLD* (K = 50) E AJUSTE DE HIPERPARÂMETROS

Domínios da experiência	**Precisão**	***Recall***	***F1-Score***
Educação/Aprendizado	79%	90%	84%
Entretenimento	75%	70%	72%
Estética/Contemplação	88%	93%	91%
Evasão/Escapismo	93%	82%	87%
MÉDIA	**84%**	**84%**	**84%**

dropout, 8 unidades na camada LSTM bidirecional, *batch size* ou tamanho de lote de 4 unidades.

D. Discussões

A métrica principal de desempenho de cada modelo usada neste trabalho foi o *F1-Score*, usando como parâmetros a precisão e o *recall*. Comparando-se a média de precisão, *recall* e o *F1-Score* para cada modelo de classificação com os melhores resultados obtidos, tem-se o resultado mostrado na tabela Tab. XVI.

Tabela XVI
MELHORES RESULTADOS OBTIDOS CADA MODELO DE CLASSIFICAÇÃO DE QUADRIGRAMAS DE ACORDO COM AS MÉTRICAS DE AVALIAÇÃO PRECISÃO, *RECALL* E *F1-SCORE*

Modelo de classificação	**Precisão**	***Recall***	***F1-Score***
Gaussian Naive Bayes	82%	81%	80%
SVM	88%	88%	88%
Bi-LSTM	84%	84%	84%

Observa-se que o melhor desempenho foi obtido pelo modelo com redes neurais do tipo SVM para classificação de quadrigramas, com *F1-Score* de 88%, se comparado com os modelos de aprendizado *Gaussian Naive Bayes* e Bi-LSTM.

De acordo com os resultados obtidos nas Tab. XIII, Tab. XIV e Tab. XV, observa-se que os modelos apresentados obtiveram resultados semelhantes, porém o modelo *Gaussian Naive Bayes* teve o menor desempenho.

De acordo com Yi Ying [45], o modelo de rede neural SVM tem uma boa capacidade de generalizar características de alta dimensão de espaços, não exigindo-se a seleção de características.

Segundo T. Joachims [46], em aprendizagem de classificadores de texto, é preciso lidar com muitas (mais de 10.000) características. Como os modelos *SVMs* usam proteção contra *overfitting*, esses não dependem necessariamente do número de características, eles têm o potencial para lidar com esses grandes espaços de características.

De acordo com T. Joachims [46], um bom classificador deve combinar muitas características e a seleção agressiva de características pode resultar em perda de informações, sendo assim os modelos *SVMs* tem poucas características irrelevantes.

T. Joachims [46] relata que os *SVMs* são adequados para problemas com conceitos densos e instâncias esparsas e que a maioria dos problemas de categorização de texto são linearmente esparsos. Contudo, a ideia de SVMs é encontrar tais separadores lineares, polinomiais, RBF, etc.

Sendo assim, os argumentos apresentados por T. Joachims [46] fornecem evidências teóricas de que os SVMs devem ter um bom desempenho para categorização de texto.

Entretanto, dentro do contexto de classificação de texto com classificador *Gaussian Naive Bayes*, foi obtido um desempenho equiparável com o resultado do modelo SVM.

Assim sendo, segundo Anderson de Rezende Rocha [47], os classificadores Naive Bayes funcionaram muito bem em muitas situações complexas do mundo real. Uma vantagem do classificador *Naive Bayes* é que ele requer apenas uma pequena quantidade de dados de treinamento para estimar os parâmetros (médias e variâncias das variáveis) necessários para a classificação. Outro fator são as distribuições de características condicionais de classe, o que significa que cada distribuição pode ser estimada independentemente como uma distribuição unidimensional.

Entretanto, a poderosa capacidade do LSTM de extrair informações avançadas de texto desempenha um papel importante na classificação de texto. O escopo de aplicação de LSTMs tem se expandido rapidamente nos últimos anos, e muitos pesquisadores propuseram muitas maneiras de renovar LSTMs para melhorar ainda mais a sua precisão [48].

Os modelos LSTM podem capturar dependências de longo prazo entre sequências de palavras, portanto, são mais usados para classificação de texto [33].

A rede neural Bi-LSTM é composta por unidades LSTM que operam em ambas as direções para incorporar informações de contexto passadas e futuras. O Bi-LSTM pode aprender dependências de longo prazo sem reter informações de contexto duplicadas [49]. Portanto, tem demonstrado excelente desempenho para problemas de modelagem sequencial e é amplamente utilizado para classificação de textos [33].

Nas figuras Fig. 19 e Fig. 20 são mostrados os gráficos de precisão e perda do modelo Bi-LSTM com sobreamostragem, validação cruzada *K-FOLD* (k = 50) e ajuste de hiperparâmetros. Na Fig. 19 é observado que a precisão em treinamento atinge quase 100% em poucas épocas de treinamento, enquanto que no cenário de validação (10% da base de dados) é obtido em torno de 60% de precisão. Verificando o gráfico de perda na Fig. 20, observa-se que a perda no treinamento do modelo Bi-LSTM é menor que na validação, também é possível observar que a partir de um ponto específico a medida que a perda no treinamento dimunui, a perda na validação aumenta drasticamemte.

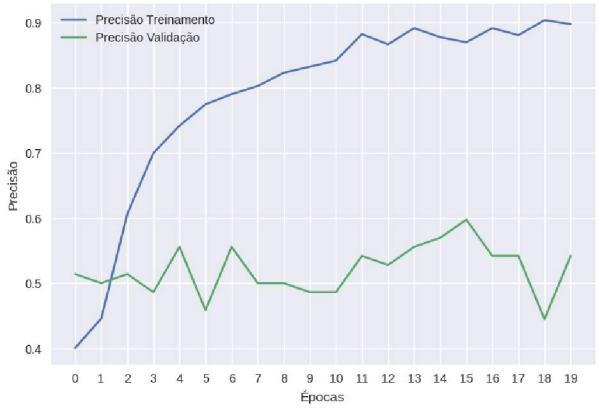


Figura 19. Gráfico de precisão do modelo de classificação Bi-LSTM em função do número de épocas de treinamento do modelo de classificação Bi-LSTM com sobreamostragem de dados, validação cruzada *K-FOLD* (k = 50) e ajuste de hiperparâmetros.

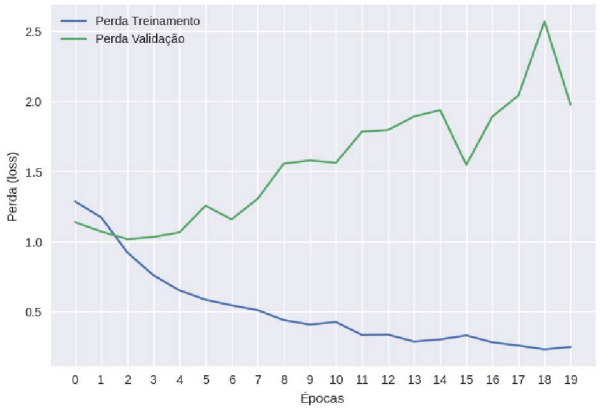


Figura 20. Gráfico de perda do modelo de classificação Bi-LSTM em função do número de épocas de treinamento do modelo de classificação Bi-LSTM com sobreamostragem de dados, validação cruzada *K-FOLD* (k = 50) e ajuste de hiperparâmetros.

De acordo com Xue Ying [50], esse comportamento se dá por uma desaceleração na velocidade de aprendizagem. Isso significa que a precisão dos algoritmos para de melhorar depois de algum ponto, ou até piora por causa do aprendizado de ruído.

Segundo Xue Ying [50], quando o modelo de rede neural não generaliza bem a partir de dados observados para dados não vistos, isso é chamado de *overfitting*. Devido à existência de *overfitting*, o modelo funciona perfeitamente no conjunto de treinamento, enquanto se ajusta mal no conjunto de teste. Isso se deve ao modelo superajustado ter dificuldades em lidar com partes das informações no conjunto de teste, o que pode ser diferente do conjunto de treinamento. Por outro lado, modelos superajustados tendem a memorizar todos os dados, incluindo ruído inevitável no conjunto de treinamento, em vez de aprender a disciplina escondida atrás dos dados.

As causas desse fenômeno podem ser complicadas. Geralmente, é possível categorizá-los em três tipos: 1) Aprendizado de ruído no conjunto de treinamento; 2) Complexidade da hipótese; e 3) Múltiplos procedimentos de comparação que são onipresentes em algoritmos de indução, bem como em outros algoritmos de inteligência [50].

Para solucionar o problema de *overfitting*, o autor Xue Ying [50] lista algumas abordagens para lidar com esse problema, que são: aplicação de *early-stopping*, redução do tamanho da rede neural, expansão dos dados de treinamento e regularização, em especial das características dos dados.

Os dados não estão em equilíbrio no que tange a quantidade de dados por classe, necessitando-se da aplicação de técnicas para balanceamento entre as classes (domínios). Como não foi possível aplicar a técnica SMOTE para o modelo Bi-LSTM, optou-se pela sobreamostragem de dados minotários da base, substituindo-os randomicamente na classe majoritária. Esse ponto certamente pode ser melhorado em trabalhos futuros, usando uma base de dados maior, com dados mais variados e devidamente balanceados entre as classes.

Como relatado por Xue Ying [50] a base de dados de treinamento pode ser expandida ao ponto que algumas abordagens podem ser tomadas: 1) Adquirir mais dados de treinamento; 2) Adicionar algum ruído aleatório ao conjunto de dados existente; 3) Recuperar alguns dados do conjunto de dados existente através de algum processamento; 4) Produzir alguns novos dados com base na distribuição do conjunto de dados existentes.

Segundo Xue Ying [50], o modelo de rede neural pode ser reduzido, pois ruídos podem estar inseridos nos dados, reduzindo o modelo de rede neural consequentemente diminuirá os ruídos inseridos na rede, que consequentemente são memorizados à medida que as camadas ocultas tratam os dados. Também é fundamental encontrar o melhor equilíbrio entre precisão e consistência, observando a variância e vieses dos dados de entrada.

De acordo com Xue Ying [50], é necessário a regularização entre o número de caracterísiticas dos dados que realmente influenciam no desempenho do modelo e a diminuição de caracterísiticas que tem menor impacto. Algumas soluções para essa regularização são citadas: 1) Selecionar apenas as características úteis e remover as características inúteis do modelo; 2) Minimizar os pesos das características que pouco influenciam na classificação final.

Contudo, observou-se uma significante melhoria de desempenho dos modelos *Gaussian Naive Bayes*, SVM e Bi-LSTM usando técnicas de sobreamostragem de dados, validação cruzada e ajuste de hiperparâmetros.

E. Trabalhos futuros

Para trabalhos futuros, primeiramente, sugere-se coletar mais dados para treinamentos e testes dos modelos. Contudo, é importante que a quantidade de dados por domínio de experiência esteja balanceada, desta forma, suprimi-se vieses dentro dos dados.

Aprender com conjuntos de dados de tamanho limitado é extremamente desafiador e, por esse motivo, em grande parte sem solução. Poucos trabalhos tentaram abordar o problema da formação de arquiteturas de aprendizado profundo com

um pequeno número de amostras devido a dificuldade de generalização para novas instâncias [51].

Não é uma boa prática usar dados desbalanceados para treinamento de modelos de redes neurais, porém o treinamento dos modelos de classificação com dados desbalanceados foram feitos para fins de comparação com outras tecnicas de inteliência artificial, avaliando-se as melhorias que essas técnicas trazem em relação ao desempenho de cada modelo em termos de precisão. Observou-se uma grande melhoria na precisão usando balanceamento de dados para classificação de textos com inteligência artificial. Entretanto, outras alternativas que podem ser testadas é a supressão dos dados minoritários, como no domínio da Evasão, que têm baixa influência na precisão de um modelo, ou estabelecer um limite de corte mínima nos dados para cada domínio de experiência, diminuindo a discrepância dos dados em termos de quantidade.

Como visto neste trabalho, a medida que novos dados são inseridos nos modelos de redes neurais, a depender do modelo de classificação de texto, o desempenho aumenta, tendo em vista as métricas apresentadas.

Outro fator que pode ser explorado é a validação cruzada, pois aumentando o número de partições, é possível observar o aumento do desempenho, porém o tempo para treinamento da rede neural aumenta consideravelmente.

O ajuste de hiperparâmetros para cada modelo também pode ser revisto, entretanto, cada modelo tem seus parâmetros, que podem ser inúmeros. Portanto, é necessário utilizar uma gama de parâmetros que são, de fato, revelantes para cada modelo.

Por fim, sugere-se o uso de outros tipos de redes neurais que se adequem melhor a natureza da classificação de quadrigramas, entre eles pode-se citar: *CNN-LSTM*, *GRU (Gate Recurrent Unit)*, *BERT (Bidirectional Encoder Representations from Transformers)*, *CNN (Convolutional Neural Network)*, algumas delas listadas por Ömer Köksal [29].

Referências

[1] O. De La Torre, "El turismo fenómeno social," México, Fondo de Cultura Económica, 1992.

[2] Brasil, "Ecoturismo: orientações básicas," 2 ed, Brasília, Ministério do Turismo, 2010, Acessado em 01 de agosto de 2022, Disponível em https://bit.ly/3rcgRjO.

[3] M. C. Beni,"Turismo: da economia de serviços à economia da experiência, Revista Turismo - Visão e Ação, p. 296-306, 2004.

[4] G. U. C. Tocantis, "Parque Estadual do Jalapão.", Tocantis, 2022, Acessado em 06 de agosto de 2022 em http://gesto.to.gov.br/uc/45/.

[5] B. J. Pine, J. H. Gilmore,"The experience economy: work is a theatre & every business a stage." Boston: Harvard Business School, 1999.

[6] E. F. Kaizer, J. E. Marynowski et al.,"Análise da Experiência relatada pelos turistas ao visitar o Parque Estadual do Jalapão (PEJ) - TO, Brasil," Ateliê do Turismo - Campo Grande/MS, UFMS, vol. 5, n.1, p. 183-204, 2021.

[7] TripAdvisor, "TripAdvisor: mais de um bilhão de avaliações de hotéis, atrações, restaurantes e muito mais," Acessado em 06 de agosto, Disponível em https://www.tripadvisor.com.br/

[8] S. Gurinder et al.,"Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification", Amity University, International Conference on Automation, Computational and Technology Management (ICACTM), 2019.

[9] O. Kennedy et al., "N-gram Based Text Categorization Method for Improved Data Mining", Journal of Information Engineering and Applications, Vol.5, No.8, 2015.

[10] Z. Wang e Z. Qu,"Research on Web Text Classification Algorithm Based on Improved CNN and SVM," School of Information Science Engineering, Lanzhou University Lanzhou, China, 17th IEEE International Conference on Communication Technology, 2017.

[11] K. Hyland,"As can be seen: lexical bundles and disciplinary variation", English for Specific Purposes, London: Elsevier, vol. 27, 2008.

[12] J. Violos, "Text Classification Using the N-Gram Graph Representation Model Over High Frequency Data Streams", Frontiers in Applied Mathematics and Statistics, Department of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece & Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece, 2018.

[13] X. Yang, "A Study of Text Vectorization Method Combining Topic Model and Transfer Learning", School of Information, Beijing Wuzi University, Beijing, China, 2022.

[14] J. L. N. Barbosa et al., "Introdução ao Processamento de Linguagem Natural usando Python", III Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos, v. 1, n. 1, p. 336-360, jun, 2017.

[15] L. A. Mullen et al.,"Fast, Consistent Tokenization of Natural Language Text", The Journal of Open Source Software, 2018.

[16] Croft, W.B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice, vol. 283. Addison-Wesley, Reading (2010).

[17] Md. H. Rahman et al., "An Investigation and Evaluation of N-Gram, TF-IDF and Ensemble Methods in Sentiment Classification", Second International Conference, ICACDS 2018, Dehradun, India, April 20-21, 2018.

[18] M. Das et al., "A Comparative Study on TF-IDF Feature Weighting Method and its Analysis using Unstructured Dataset", NIT Trichy, 620015, Tamil Nadu, India, 2020.

[19] F. Heimerl, M. Gleicher, "Interactive Analysis of Word Vector Embeddings", Computer Graphics Forum, 37(3), 253–265. https://doi.org/10.1111/cgf.13417, 2018.

[20] S. Yadav e S. Shukla,"Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," IEEE 6th International Conference on Advanced Computing, 2016.

[21] E. B. Rabello, "Cross Validation: Avaliando seu modelo de Machine Learning", Acessado em 01 de agosto de 2022, Disponível em https://medium.com/@edubrazrabello/cross-validation-avaliando-seu-modelo-de-machine-learning-1fb70df15b78.

[22] N. V. Chawla, "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research 16(1):321-357, June 2002.

[23] A. Bernardo et al., "C-SMOTE: Continuous Synthetic Minority Oversampling for Evolving Data Streams", Politecnico di Milano, Dipartimento di Elettronica Informazione e Bioingegneria, Dec. 2020.

[24] Y. Liu et al., "An Improvement of One-against-all Method for Multiclass Support Vector Machine", International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, 2007.

[25] Baeldung, "Multiclass Classification Using Support Vector Machines", August 2021, Acessado em 01 de agosto de 2022, Disponível em https://www.baeldung.com/cs/svm-multiclass-classification.

[26] Venkatesh,"Classification and Optimization Scheme for Text Data using Machine Learning Naïve Bayes Classifier," University Visvesvaraya College of Engineering, World Symposium on Communication Engineering, 2018.

[27] M. Hammond, "Statistical Natural Language Processing", University of the Philippines, Chapter 4: N-grams, Acessado em 16 de outubro de 2022, Disponível em https://faculty.sbs.arizona.edu/hammond/archive/ling696f-sp03/snlp4.pdf.

[28] D. Pinto, David Pinto, "Bayesian Classification with Regularized Gaussian Models", Belo Horizonte - MG, Nov. 2015.

[29] O. Köksal,"A Comparative Text Classification Study with Deep Learning-Based Algorithms,"Artificial Intelligence and Information Technologies Department ASELSAN, 9th International Conference on Electrical and Electronics Engineering, 2022.

[30] S. Hochreiter e J. Schmidhuber,"Long Short-Term Memory," Neural Comput.," vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[31] Prof. Dr. R. A. N. R. Montaño, "RNN e Classificacao de Textos," Notas de aula: Frameworks de IA, Acessado em 06 de agosto de 2022, Disponível em https://www.youtube.com/watch?v=b9lOy5dE6jg.

[32] X. Peng, "A Comparative Study of Neural Network for Text Classification," College of Mathematics, Jilin University, 2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), 2020.

[33] B. Jang et al., "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism", Applied Sciences, August 2020.
[34] Z. Cui et al., "Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction", Transportation Research Part C Emerging Technologies, September 2020.
[35] L. Yang, A. Shami, "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice", Department of Electrical and Computer Engineering, University of Western Ontario, 2020.
[36] C. Manliguez, "Generalized Confusion Matrix for Multiples Classes", University of the Philippines, Nov. 2016.
[37] Z. C. Lipton, "Thresholding Classifiers to Maximize F1 Score", University of California, San Diego, USA, 2014.
[38] A. Kayid et al. "Performance of CPUs/GPUs for Deep Learning workloads", Media Engineering and Technology Faculty, German University in Cairo, May 2018.
[39] M. Ashfaq. "An Introduction to Deep Convolutional Neural Networks With Keras", Chandigarh University, Feb 2021.
[40] D. Makienko et al., "The effect of the imbalanced training dataset on the quality of classification of lithotypes via whole core photos", VI International Conference on Information Technology and Nanotechnology, 2020.
[41] S. Kotsiantis, "Handling imbalanced datasets: A review", GESTS International Transactions on Computer Science and Engineering, Vol.30, 2006.
[42] E. Elgeldawi, "Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis", Informatics, 2021.
[43] J. N. v. Rijn, "Hyperparameter Importance Across Datasets", Albert-Ludwigs-Universität Freiburg, London, United Kingdom, 2018.
[44] N. Luaran, "Assessment of the Optimization of Hyperparameters in Deep LSTM for Time Series Sea Water Tidal Shift", Universiti Malaysia Sabah, Research Square, https://doi.org/10.21203/rs.3.rs-1669035/v1, 2022.
[45] Y. Ying, "Effectiveness of the News Text Classification Test Using the Naïve Bayes' Classification Text Mining Method", Journal of Physics: Conference Series, 2021.
[46] T. Joachims, "Text Categorization with Supp ort Vector Machines: Learning with Many Relevant Features", Cornell University, January 1998.
[47] A. de R. Rocha, "Notas de Aula: Naive Bayes classifier", Institute of Computing (Unicamp), SP, Brazil, 2011.
[48] A. S. Graves, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. 2005, 18, 602–610.
[49] D. Liang, Y. Zhang, AC-BLSTM: Asymmetric convolutional bidirectional LSTM networks for text classification. arXiv 2016, arXiv:1611.01884.
[50] X. Ying, "An Overview of Overfitting and its Solutions", Journal of Physics: Conference Series, 2019.
[51] L. Brigato et al., "A Close Look at Deep Learning with Small Data", Dpt. of Computer, Control, and Management Engineering, Sapienza University of Rome, Rome, Italy, 2020.