



**UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACULDADE DE COMPUTAÇÃO**

**Inteligência Artificial
Edson Takashi Matsubara**

MAYKON KAZUHIRO FALCÃO TAMANAHA

RELATÓRIO TRABALHO 1

**Classificação Automática de Escolas de Magia em D&D
usando Machine Learning e Sentence Embeddings**

**CAMPO GRANDE - MS
06/10/2025**



1 INTRODUÇÃO

O aprendizado de máquina, ramo da Inteligência Artificial voltado ao treinamento de sistemas computacionais para a execução de tarefas específicas, foi empregado neste trabalho com foco na técnica de aprendizagem supervisionada. O objetivo consistiu em classificar e rotular textos, tendo como entrada uma descrição textual e como saída a respectiva classe.

Para o treinamento do modelo, os dados foram obtidos por meio de Web Scraping, técnica de mineração voltada à extração automatizada de informações de páginas da web. Foram coletados dados referentes ao nome, à categoria e à descrição dos elementos, extraídos diretamente do site, posteriormente processados e empregados no treinamento do modelo.

O processo incluiu a utilização de embeddings, representações vetoriais densas que permitem converter textos em formatos numéricos, preservando relações semânticas entre palavras e frases. Esses embeddings foram gerados a partir das descrições dos feitiços, servindo como entrada para o classificador. Para garantir desempenho consistente e evitar sobreajuste, aplicou-se validação cruzada (cross-validation), técnica estatística que consiste em particionar o conjunto de dados em subconjuntos (folds) para treinar e avaliar o modelo em diferentes segmentos, assegurando que os resultados sejam generalizáveis.

O tema escolhido para o trabalho foi Dungeons & Dragons, em que o modelo recebeu como entrada a descrição de um feitiço e retornou a escola de magia à qual ele pertence. As escolas consideradas foram: abjuration, conjuration, divination, enchantment, evocation, illusion, necromancy e transmutation. Assim, o modelo pôde mapear, por meio dos embeddings, a relação semântica entre a descrição e as classes, sendo avaliado de forma robusta por meio da validação cruzada.

As métricas para avaliação foram:

Precisão, indica a proporção de previsões positivas que realmente são corretas.

$$\text{Precisão} = \frac{VP}{VP + FP}$$

Recall, mede a proporção de exemplos positivos que foram corretamente identificados pelo modelo.

$$\text{Recall} = \frac{VP}{VP + FN}$$

F1-Score é a média harmônica entre precisão e recall, oferecendo um equilíbrio entre ambas as métricas, especialmente útil quando há desequilíbrio entre classes.

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$$

2 OBJETIVOS

O objetivo é classificar automaticamente feitiços de Dungeons & Dragons (D&D) em suas respectivas escolas de magia baseando-se apenas nas descrições textuais dos feitiços. As escolas de magia consideradas foram Abjuration, focada em proteção, defesa e anulação de efeitos mágicos adversos; Necromancy, relacionada à manipulação da energia da morte, vida e mortos-vivos; Evocation, que cria e manipula energia para causar dano direto, como fogo, eletricidade ou explosões; Transmutation, que altera as propriedades físicas ou químicas de objetos ou criaturas; Illusion, que cria imagens, sons ou sensações falsas para enganar os sentidos; Enchantment, que influencia a mente e o comportamento de outras criaturas; Divination, que revela informações ocultas ou futuras, como previsões e detecção; e Conjunction, que invoca criaturas, objetos ou transporta instantaneamente seres e coisas.

A primeira parte é o web scraping do site <https://www.dndbeyond.com/spells?page=1>, seguida do processamento dos dados para remover feitiços vazios e corrigir colunas do arquivo CSV, assim como a preparação do dataset com divisão em conjuntos de treinamento, validação e teste, além da realização do embedding. Por fim, o treinamento e avaliação do modelo utilizando validação cruzada e métricas de precisão, recall e f1-score.

3 MÉTODOS

A coleta de dados foi realizada através de web scraping do site D&D Beyond, extraindo informações de 46 páginas de feitiços. O arquivo resultante (`feiticos.csv`) contém três campos: nome do feitiço, escola de magia e descrição completa.

O pré-processamento incluiu correção de inconsistências no arquivo CSV, identificando e corrigindo linhas com número incorreto de campos, vírgulas não escapadas.

A separação do dataset foi feita de forma estratificada para manter a proporção de classes em todos os conjuntos, de acordo com a relação Treino: 70%, Validação: 20%, Teste: 10%, utilizando o parâmetro `stratify` do `sklearn.train_test_split` para garantir representatividade balanceada das oito escolas de magia.

Para a geração de embeddings, foi escolhido o modelo `all-MiniLM-L6-v2` da biblioteca Sentence Transformers, por suas capacidades de semântica contextual (embeddings de 384 dimensões) e aplicabilidade a textos descritivos, como os de feitiços de D&D.

Os embeddings foram armazenados em formato comprimido (`.npz`) para otimizar o espaço em disco e reduzir o tempo de carregamento durante os experimentos posteriores.

3.1 OTIMIZAÇÃO DE HIPERPARÂMETROS

3.1.1 ANÁLISE DOS DADOS

O conjunto original de dados (*feiticos.csv*) continha 905 registros, mas apresentava inconsistências de formatação, como linhas com número incorreto de campos, o que exigiu um processo de correção e limpeza. Após o pré-processamento, o arquivo resultante (*feiticos_fixed.csv*) passou a contar com 700 registros válidos, indicando a remoção de aproximadamente 22,7% dos dados devido a erros estruturais. A distribuição das classes (escolas de magia) manteve-se relativamente proporcional, destacando-se as categorias *transmutation* (20%), *evocation* (18,6%) e *conjunction* (15,3%), enquanto *necromancy* (7,6%), *illusion* (8%) e *divination* (8,7%) apresentaram menor representatividade, refletindo um leve desbalanceamento entre as classes. A divisão dos dados foi realizada em conjuntos de treino (490 registros, 70%), validação (140 registros, 20%) e teste (70 registros, 10%), com estratificação, de modo a preservar a proporção das classes e evitar viés no aprendizado do modelo. A análise comparativa mostrou que a maior parte dos registros removidos pertenceu às classes *conjunction* (-37,1%), *enchantment* (-30%) e *evocation* (-24,4%), o que pode afetar o desempenho nessas categorias. O pré-processamento incluiu a remoção de linhas com inconsistências, correção de campos e estratificação dos dados, garantindo um conjunto limpo, coerente e representativo para o treinamento. Após a limpeza, não foram identificados valores nulos, assegurando uma base consistente e adequada para o processo de modelagem.

3.1.2 NORMALIZAÇÃO DOS DADOS

Todos os modelos foram treinados com embeddings normalizados usando `normalize` do `sklearn.preprocessing`:

$$x_{\text{norm}} = \frac{x}{\|x\|} \quad (1)$$

onde $\|x\|$ representa a norma (por exemplo, L2) do vetor de embeddings, calculada para cada amostra individualmente.

3.1.3 VALIDAÇÃO CRUZADA

Implementou-se validação cruzada estratificada com $k = 5$, escolhido por conveniência, folds para avaliação robusta dos hiperparâmetros. Os conjuntos de treino e validação foram combinados para maximizar os dados disponíveis, enquanto o conjunto de teste manteve-se separado para a validação final.

3.1.4 HIPERPARÂMETROS TESTADOS

A Tabela 1 apresenta todos os hiperparâmetros avaliados via grid search:

Tabela 1 – Hiperparâmetros testados via Grid Search com validação cruzada

Modelo	Hiperparâmetro	Valores Testados
KNN	n_neighbors metric weights	[1, 3, 5, 7, ..., 29] (15 valores) [euclidean, manhattan, cosine, minkowski, hamming] distance (fixo)
Regressão Logística	C penalty	[0.001, 0.01, 0.1, 1, 10, 100, 1000] l2 (fixo)
SVM	C gamma kernel	[0.01, 0.1, 1, 10, 100, 1000] [0.0001, 0.001, 0.01, 0.1, 1, 10] rbf (fixo)

3.1.5 MÉTRICA DE AVALIAÇÃO

Foi utilizado o **F1-score weighted** como métrica principal para seleção de hiperparâmetros:

$$F1_{weighted} = \sum_{i=1}^C w_i \cdot F1_i \quad (2)$$

onde w_i é a proporção de exemplos da classe i e C é o número de classes. Esta métrica foi escolhida por considerar tanto precisão quanto recall e ponderar adequadamente classes com diferentes frequências.

3.1.6 PROCESSO DE OTIMIZAÇÃO

Após identificar os melhores hiperparâmetros, o modelo final foi treinado com todos os dados de treino+validação e avaliado no conjunto de teste separado.

3.2 MODELOS DE CLASSIFICAÇÃO

Os modelos de classificação utilizados foram K-Nearest Neighbors (KNN), Regressão Logística e Support Vector Machine (SVM). O KNN foi escolhido como baseline não-paramétrico por não assumir distribuição específica dos dados e ser eficaz em espaços de embeddings bem estruturados, realizando a classificação com base na vizinhança local, sem exigir fase de treinamento explícita, sendo sensível à métrica de distância e utilizando `weights='distance'` para ponderar vizinhos mais próximos. A Regressão Logística, um modelo linear simples e eficaz, foi empregada para comparar se a complexidade de modelos não-lineares é necessária, realizando classificação multiclasse via one-vs-rest, com regularização L_2 (Ridge) controlada pelo parâmetro

C e permitindo interpretabilidade dos pesos aprendidos. Já a SVM foi selecionada por criar fronteiras de decisão complexas em espaços de alta dimensionalidade por meio do kernel trick, sendo robusta e eficaz em classificação multiclasse, utilizando kernel RBF ($K(x, x') = \exp(-\gamma\|x - x'\|^2)$), maximizando a margem entre classes, regularizada pelo parâmetro C e sensível à escala dos dados, o que exige normalização.

4 RESULTADOS E DISCUSSÕES

4.1 HIPERPARÂMETROS ÓTIMOS ENCONTRADOS

A Tabela 2 apresenta os melhores hiperparâmetros identificados para cada modelo:

Tabela 2 – Melhores hiperparâmetros encontrados via validação cruzada

Modelo	Hiperparâmetro	Valor Ótimo
KNN	K	1
	Métrica	manhattan
Regressão Logística	C	10
SVM	C	10
	Gamma	1

4.2 COMPARAÇÃO DE DESEMPENHO

A Tabela 3 apresenta as métricas comparativas dos três modelos no conjunto de teste:

Tabela 3 – Comparação de desempenho usando média ponderada dos modelos no conjunto de teste

Modelo	Acurácia	Precisão	Recall	F1-Score
KNN	0.87	0.88	0.87	0.87
Regressão Logística	0.81	0.83	0.81	0.81
SVM	0.84	0.86	0.84	0.84

5 CONCLUSÕES

Este trabalho demonstrou a viabilidade de classificar automaticamente feitiços de D&D em suas escolas de magia utilizando técnicas modernas de NLP e algoritmos clássicos de machine learning, dispensando a necessidade de feature engineering manual extensivo. A estratégia baseada em Sentence Embeddings combinada com modelos tradicionais mostrou-se eficaz e adaptável, viabilizando um pipeline robusto, modular e reproduzível, adequado para futuras extensões.

No processo de avaliação, a validação cruzada e a otimização dos hiperparâmetros garantiram justiça e evitaram vieses experimentais. Entre os três modelos analisados,

o KNN apresentou desempenho superior em todas as métricas principais (acurácia, precisão, recall e F1-score), enquanto SVM e Regressão Logística tiveram resultados próximos, porém inferiores. Assim, a escolha do modelo ideal deve considerar o equilíbrio desejado entre acurácia, interpretabilidade e eficiência computacional no contexto específico de aplicação.

REFERÊNCIAS

1. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.
2. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
3. Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21-27.
4. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.
5. Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (Vol. 398). John Wiley & Sons.
6. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai*, 14(2), 1137-1145.