Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Informática
Programa de Especialização em *Data Science* e *Big Data*

Gabriel Pereira

Avaliação de classificadores de texto para verbatins de clientes do ramo de previdência utilizando aprendizado de máquina e LLMs

Curitiba
2025

Gabriel Pereira

Avaliação de classificadores de texto para verbatins de clientes do ramo de previdência utilizando aprendizado de máquina e LLMs

Monografia apresentada ao Programa de Especialização em *Data Science* e *Big Data* da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Prof. Luiz Eduardo Soares de Oliveira

Curitiba
2025

Avaliação de classificadores de texto para verbatins de clientes do ramo de previdência utilizando aprendizado de máquina e LLM

Evaluation of text classifiers for pension plan client verbatims using machine learning and LLM

Gabriel Pereira[1], Luiz Eduardo Soares de Oliveira[2]

[1]Aluno do programa de Especialização em Data Science & Big Data, gabrielpereira6671@gmail.com
[2]Professor do Departamento de Informática - UFPR, luiz.oliveira@ufpr.br

Este trabalho avalia o desempenho de modelos de aprendizado de máquina supervisionado e explora o potencial de Modelos de Linguagem Grandes (LLMs) para a classificação automática de verbatins de clientes do ramo de previdência. Foram comparados os seguintes modelos: Naive Bayes, Regressão Logística e Random Forest, utilizando vetorizações TF-IDF e embeddings BERT em dois cenários, um com 48 classes e outro com as 10 mais frequentes. A análise inicial apresentou um desbalanceamento, resultando em desempenhos baixos. Testes exploratórios com um LLM (Mistral) confirmaram a hipótese de que a dificuldade dos modelos estava ligada a desafios na taxonomia das categorias e na qualidade dos rótulos. Para validar está hipótese, a taxonomia foi refinada, resultando em um novo dataset com 5 classes. Ao re-treinar o melhor modelo, Random Forest com TF-IDF, neste novo cenário, o desempenho melhorou.
Palavras-chave: Classificação, verbatins, taxonomia, Random Forest, classes, desempenho, aprendizado de máquina, qualidade de dados

This work evaluates the performance of supervised machine learning models and explores the potential of Large Language Models (LLMs) for the automatic classification of client verbatims in the pension plan industry. The following models were compared: Naive Bayes, Logistic Regression, and Random Forest, using TF-IDF vectorization and BERT embeddings in two scenarios, one with 48 classes and another with the 10 most frequent ones. The initial analysis revealed a class imbalance, resulting in low performance. Exploratory tests with an LLM (Mistral) confirmed the hypothesis that the models' difficulty was linked to challenges in the category taxonomy and label quality. To validate this hypothesis, the taxonomy was refined, resulting in a new dataset with 5 classes. Upon retraining the best model, Random Forest with TF-IDF, in this new scenario, the performance improved.
Keywords: Classification, verbatims, taxonomy, Random Forest, classes, performance, machine learning, data quality

1. Introdução

A análise de feedback de clientes é uma ferramenta necessária para empresas que buscam aprimorar seus serviços e produtos, especialmente em setores competitivos e regulados como o de previdência privada, no ramo de seguros. A voz do cliente busca transmitir a satisfação dele através de diferentes canais, como: pesquisas de satisfação (ex: NPS), redes sociais, sites de reclamações e centrais de atendimento. E para utilizar essas informações de forma estratégica é necessário um tratamento desses dados, e um exemplo é o processo de categorização de comentários dos clientes, também conhecido como "verbatins". A classificação de texto permite visualizar os principais tópicos e a frequência de cada um, determinando assim o que a empresa deve priorizar quando o assunto é centralidade do cliente.

Esses insumos contêm informações valiosas para o direcionamento das áreas internas, contudo empresas com grande volume de clientes podem não conseguir abranger todas as fontes de dados e processá-las em tempo hábil, pois esse processo requer uma análise manual. Neste contexto, o Processamento de Linguagem Natural (PLN) e as técnicas de Machine Learning (ML) surgem como possíveis soluções para automatizar em alguma medida a análise desses insumos.

Este Trabalho de Conclusão de Curso aborda como modelos de classificação de texto podem automatizar o processo de categorização de verbatins de clientes

que possuem um seguro previdência em uma empresa de seguros. O objetivo principal é construir e comparar sistemas capazes de categorizar automaticamente as respostas textuais dos clientes em tópicos mais relevantes para o negócio, explorando tanto abordagens supervisionadas quanto o potencial de Modelos de Linguagem Grandes.

2. Discussão

Para a tarefa de classificação de texto, existem inúmeras opções de modelos supervisionados, para esse estudo foram selecionados e avaliados diferentes algoritmos. A escolha buscou contemplar modelos com distintas características, desde abordagens lineares mais simples e classificadores probabilísticos até métodos de conjunto mais complexos. O objetivo desta seleção foi permitir uma análise comparativa das capacidades de aprendizado e generalização de cada técnica para o problema atual de categorização de texto. Os modelos supervisionados investigados neste estudo foram o Naive Bayes, a Regressão Logística e o Random Forest.

O classificador Naive Bayes é um modelo probabilístico baseado no Teorema de Bayes, que opera sob uma forte (ou "ingênua") suposição de independência condicional entre as features (neste caso, as palavras ou termos do texto) dado o valor da variável de classe [1]. Apesar dessa suposição simplificadora, que raramente se sustenta em dados do mundo real, o Naive Bayes é frequentemente utilizado como um baseline eficaz em tarefas de classificação de texto devido à sua simplicidade, rapidez de treinamento e bom desempenho em muitos cenários, especialmente com dados de alta dimensionalidade. A escolha da variante do Naive Bayes depende da natureza das features de entrada. Neste trabalho, foram exploradas duas de suas principais variantes para se adequar às diferentes técnicas de vetorização utilizadas:

- **Multinomial Naive Bayes (MNB)**: Adequado para features que representam contagens discretas, como as frequências de palavras geradas pela vetorização TF-IDF.
- **Gaussian Naive Bayes (GNB):** Projetado para features contínuas, assumindo que estas seguem uma distribuição gaussiana (normal) dentro de cada classe, sendo assim aplicado sobre os embeddings BERT.

A Regressão Logística é um modelo linear generalizado amplamente empregado para problemas de classificação [2]. Embora seja um modelo linear, ele utiliza uma função logística (sigmoide para classificação binária ou softmax para multiclasse) para transformar a saída linear em probabilidades de pertencimento a cada classe. É um modelo de caráter paramétrico, que busca encontrar uma relação linear (transformada) entre as variáveis preditoras (features do texto) e o logit da probabilidade da classe de resposta. Suas vantagens incluem relativa interpretabilidade dos coeficientes (que podem indicar a importância das features), eficiência computacional e robustez em diversos tipos de dados.

O Random Forest pertence à família de algoritmos de ensemble learning, que funcionam combinando as predições de múltiplos modelos mais simples para obter um resultado final mais robusto e preciso. Especificamente, o Random Forest constrói um "bosque"composto por um grande número de árvores de decisão individuais. Cada árvore é treinada em uma amostra aleatória dos dados de treinamento (selecionada com reposição, técnica conhecida como bagging) e, em cada nó durante a construção da árvore, apenas um subconjunto aleatório de features é considerado para determinar a melhor divisão. Esta dupla aleatoriedade ajuda a criar árvores diversificadas e a reduzir a variância do modelo final, tornando-o menos propenso a overfitting em comparação com uma única árvore de decisão. Para tarefas de classificação, a predição final do Random Forest é tipicamente determinada por uma votação majoritária entre todas as árvores do bosque [3]. O Random Forest é reconhecido por sua capacidade de capturar relações não lineares e interações complexas entre as variáveis.

Modelos de aprendizado de máquina não operam diretamente sobre texto bruto; eles precisam de uma representação numérica dos dados. O processo de converter texto em vetores numéricos é conhecido como vetorização ou extração de features. A escolha da técnica de vetorização define como o significado e a estrutura do texto são apresentados ao modelo. Foram exploradas duas abordagens distintas e amplamente utilizadas: a técnica estatística TF-IDF e os embeddings contextuais gerados pelo modelo BERT.

A técnica TF-IDF é um método estatístico que converte texto em vetores numéricos, atribuindo um peso a cada palavra. O peso de uma palavra aumenta se ela aparece muitas vezes em um documento, mas diminui se ela é muito comum em todos os documentos do conjunto. Dessa forma, palavras que são específicas e importantes para um determinado documento recebem um peso maior. O resultado é um vetor numérico

para cada texto, onde cada posição representa uma palavra do vocabulário. Mesmo que seja uma técnica rápida e eficiente, o TF-IDF não captura o significado ou o contexto das palavras, tratando-as de forma isolada.

Em contraste com o TF-IDF, os embeddings BERT buscam capturar o significado semântico e o contexto das palavras em uma frase. BERT é um modelo pré-treinado em um vasto volume de textos que analisa as palavras em relação a todas as outras na sentença, de forma bidirecional. Neste estudo, o BERT foi utilizado para transformar cada verbatim em um vetor denso de 768 dimensões, chamado de embedding. Este vetor representa o significado da sentença como um todo, posicionando textos com significados similares próximos no espaço vetorial. A principal vantagem é a capacidade de aprender padrões semânticos complexos, enquanto a desvantagem é o maior custo computacional para gerar esses vetores.

Adicionalmente às técnicas de aprendizado supervisionado descritas anteriormente, que requerem um conjunto de dados já categorizado para treinamento específico no problema, também foi investigado de forma qualitativa como os Modelos de Linguagem Grandes (LLMs) podem auxiliar o processo. LLMs, como o mistral e Qwen utilizados neste estudo (acessado localmente via Ollama ou por python), são modelos pré-treinados em volumes massivos e diversificados de dados textuais, o que lhes confere um vasto conhecimento linguístico e uma capacidade intrínseca de compreender e gerar linguagem natural [4]

Para a avaliação do desempenho de modelos de classificação é necessário o uso de um conjunto de métricas. As métrica utilizadas para esse estudo foram, acurácia geral, precisão e recall, F1-score e F1-score Weighted.

A acurácia Geral representa a proporção de todas as predições que o modelo acertou. Embora intuitiva, em datasets desbalanceados, um modelo pode alcançar alta acurácia simplesmente ao prever a classe majoritária, mascarando um desempenho ruim nas classes minoritárias.

A precisão e recall (Sensibilidade) são métricas calculadas por classe. A Precisão mede, de tudo que foi classificado como uma classe X, quanto estava correto. O Recall mede, de tudo que realmente era da classe X, quanto o modelo conseguiu identificar. Há um troca entre elas, sendo aumentar o recall pode diminuir a precisão, e vice-versa.

A F1-score macro calcula o F1-score para cada classe individualmente e depois tira a média aritmética simples. Esta métrica trata todas as classes com igual importância, independentemente do número de amostras em cada uma. Por essa razão, o F1-score Macro é importante para avaliar o desempenho em datasets desbalanceados, pois um desempenho ruim em classes minoritárias penaliza fortemente a média geral.

F1-score Weighted: Também calcula a média dos F1-scores por classe, mas de forma ponderada pelo número de amostras de cada classe. Ele dá mais peso às classes maiores e tende a ser numericamente próximo da acurácia geral.

Matriz de Confusão: É uma tabela que visualiza o desempenho do classificador, mostrando os acertos na diagonal principal e os erros de classificação fora dela. É uma ferramenta essencial para a análise qualitativa dos erros do modelo.

3. Materiais e métodos

A base de dados utilizada neste trabalho foi fornecida pela empresa contendo 5232 registros (linhas) de feedback de clientes de previdência. A base original incluía diversas colunas com diferentes dados, e para esse aprofundamento as colunas utilizadas foram "Motivo da nota", que continha o texto do verbatim na coluna e as colunas "Tópico PAI 1", "Tópico FILHO 1", "Tópico PAI 2", "Tópico FILHO 2"que eram preenchidas pelo analista, vale destacar que a coluna "Tópico PAI"representa a categoria principal e "Tópico FILHO"a subcategoria.

Um desafio inicial é a presença de múltiplas colunas de categorização e valores nulos e a partir disso foi feito uma curadoria manual, com o objetivo de criar um rótulo único para cada verbatim, a chamada coluna "Tópico Final 1". O primeiro passo foi remoção de valores nulos e posteriormente definir qual dado estará presente na coluna "Tópico Final 1", e para isso foi concatenado o topico pai com o filho, dado que são complementares e foi feita a priorização, como a utilização preferencial da hierarquia "Tópico PAI/FILHO 1"sobre a "Tópico PAI/FILHO 2", quando ambas existiam, e caso o "Tópico PAI/FILHO 1"estivesse em branco era considerado o "Tópico PAI/FILHO 2". Resultando em um arquivo contendo aproximadamente 1430 registros.

Após esse tratamento inicial foi feito o pré-processamento textual da coluna "Motivo da nota", utilizando a biblioteca NTLK. As etapas deste incluíram: conversão de

todo o texto para letras minúsculas, remoção completa de sinais de pontuação, tokenização em nível de palavra e a remoção de stopwords, com base na lista padrão do NLTK para o português. A coluna categoria_alvo, que representa a variável dependente para os modelos de classificação, foi definida como uma cópia direta da coluna "Tópico Final 1". Posteriormente, quaisquer linhas com valores nulos nas colunas texto_processado ou categoria_alvo foram excluídas. Este pipeline de pré-processamento resultou em um arquivo contendo 1424 registros prontos para as etapas subsequentes. O processo pode ser conferido na Figura 1

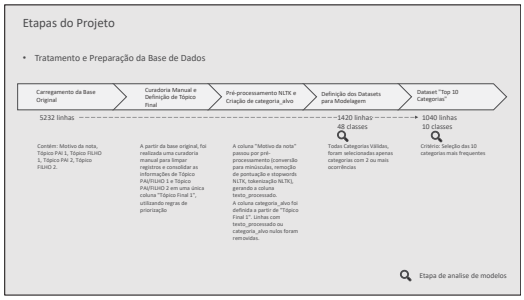

Figura 1: Processo para tratamento do dataset

A partir desta base tratada, foram definidos dois cenários de datasets para a modelagem supervisionada. O primeiro cenário, chamado "Todas Categorias Válidas", foi criado selecionando apenas categorias com um mínimo de duas ocorrências, resultando em um dataset com 1420 registros em 48 classes. O segundo, chamado "Top 10 Categorias", focou nas dez categorias mais frequentes para obter um conjunto mais balanceado, consolidando 1040 registros. A criação deste segundo cenário foi motivada pela baixa performance observada nos testes iniciais com 48 classes. Para ambos os cenários, os datasets foram divididos em conjuntos de treinamento (80%) e teste (20%) de forma estratificada e com semente aleatória. O vetorizador é treinado uma única vez com o conjunto de dados mais amplo ("Todas as Categorias Válidas"). Depois, ele é usado para transformar tanto o conjunto "Todas as Categorias"quanto o "Top 10 Categorias". Isso garante que ambos os modelos operem no mesmo espaço de features, tornando a comparação metodologicamente correta.

Para a vetorização dos textos pré-processados, duas técnicas foram empregadas: TF-IDF (Term Frequency-Inverse Document Frequency), um método estatístico baseado em frequência de palavras, e embeddings contextuais gerados pelo modelo BERT, que convertem textos em vetores densos que capturam significado semântico.

Os modelos de classificação supervisionada avaliados neste estudo cada um foi treinado e testado nos cenários "Todas Categorias Válidas"e "Top 10 Categorias".

O desempenho de todos os modelos foi avaliado no conjunto de teste utilizando um conjunto de métricas padrão: Acurácia Geral, Recall e F1-score, calculados por classe e agregados (Macro e Weighted). Para visualização dos padrões de erro, foram geradas Matrizes de Confusão. Todo o pipeline computacional foi implementado em Python (versão 3.x), com o apoio de bibliotecas como Pandas, NumPy, Scikit-learn, Transformers, Sentence Transformers e Matplotlib/Seaborn.

Além dos testes com modelo supervisionado, foram realizandos testes qualitativos com LLM. O primeiro teste feito foi com o Mistral via Ollama local, para a realização dele foi coletado uma amostra de 25 verbatims, 5 de cada uma das 5 categorias mais frequentes. Outro teste realizado foi com o Qwen 1.5, executado localmente por meio da plataforma Ollama, com suporte via VS Code e ambiente Python. Para este experimento, foram selecionadas as 10 primeiras linhas da planilha, contendo verbatins associados às 5 categorias mais frequentes da base.

Para os testes o texto foi enviado aos modelos utilizando um prompt controlado, instruindo os modelos a escolher a categoria mais adequada a partir de determinados verbatins, um exemplo do prompt utilizado foi:

Você é um classificador de comentários. Responda apenas com a categoria mais adequada.
Classifique o seguinte texto: "[verbatim]"
Categorias:

- *Atendimento – Qualidade do atendimento*
- *Investimento – Rendimento do valor*
- *Processo – Clareza / Informação*
- *Processo – Tramites*
- *Produto – Proposta de Valor*

4. Resultados

Para avaliar os resultados foram extraídas as métricas selecionadas para os modelos supervisionados e comparadas entre modelos para definir qual teve o melhor desempenho. Para essa avaliação foram selecionados os dados do cenário de "Top 10 categorias"e apresentadas na Figura 2

MODELO (10 classes)	ACURACIA	F1-SCORE MACRO	F1-score Weighted
Naive Bayes + TF-IDF	53,85%	24%	45,00%
Gaussian Naive Bayes + BERT	57,21%	43%	59,00%
Regressão Logística + TF-IDF	66,35%	39%	60,00%
Regressão Logística + BERT	64,90%	46%	63,00%
Random Forest + TF-IDF	70,67%	47%	69%
Random Forest + BERT	46,63%	21%	38%

Figura 2: Tabela com resultados dos modelos supervisionados no cenário de Top 10 categorias

Selecionar as Top 10 categorias resultou em uma melhora no desempenho de todos os modelos supervisionados. O Random Forest com TF-IDF foi o modelo com maior desempenho em todas as métricas, com acurácia geral de 70%, F1-score Macro de 47% e F1-score Weighted de 69%. A Regressão Logística com embeddings BERT também apresentou um desempenho competitivo e equilibrado, com o segundo maior F1-score Macro 46,00% e uma acurácia de 64,90%, superando a Regressão Logística com TF-IDF anotando a F1-Macro de 39,00% e Acurácia 66,35%, em termos de balanceamento, apesar de uma ligeira desvantagem na acurácia. As abordagens com Naive Bayesa presentaram resultados inferiores em comparação com os modelos de Regressão Logística e Random Forest neste cenário. Como o modelo Random Forest com TF-IDF alcançou o melhor desempenho em todas as métricas, ele foi objeto de análise. E a partir dela foi identificado um alto desbalanceamento das classes, conforme a matriz de confusão na Figura 3 e a métrica de F1-score Macro. Vale ressaltar que esse desbalanceamento foi visto em todos os modelos.

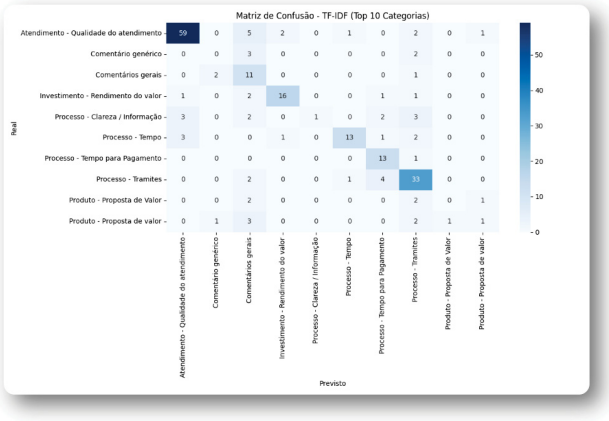

Figura 3: Matriz de confusão do Random Forest com Tf-IDF no cenário de 10 categorias.

A partir dos resultados com Random Forest com TF-IDF uma hipótese inicial do desbalanceamento classes é a quantidade de itens, essa quantidade pode ser conferida na Figura 4

Categoria_alvo	Quantidade ocorrência	RF + TD-IDF | Recall
Atendimento - Qualidade do atendimento	350	84%
Processo - Tramites	198	82%
Investimento - Rendimento do valor	102	76%
Processo - Tempo	97	65%
Comentários gerais	72	79%
Processo - Tempo para Pagamento	69	93%
Processo - Clareza / Informação	56	9%
Produto - Proposta de valor	39	12%
Comentário genérico	28	0%
Produto - Proposta de Valor	27	0%

Figura 4: Tabela com quantidade de itens das classes nos testes

As classes 'Atendimento - Qualidade do atendimento' com 84% de recall e 'Processo - Tramites' com 82% de recall são as mais volumosas se comparada as outras classes e têm um alto recall. Ao analisar outras categorias, de fato identifica-se que classes com menor volume têm um recall menor. Porém isso não é um padrão, algumas classes com menos volume também têm um recall alto, por exemplo: "Processo - Tempo para Pagamento"tem 93% de recall, inclusive com valores maiores que as exemplificadas. Logo a quantidade de itens por classe não está totalmente relacionada ao desempenho.

Como o modelo ainda apresentava desafios, principalmente classes semanticamente parecidas e em uma medida menor o baixo volume, uma análise qualitativa com LLMs foi realizada para investigar com mais detalhes o impacto dos rotulos das categorias. E nessa investigação a hipótese da qualidade da taxonomia se reforçou, e a partir dela os resultados obtidos com o Mistral foram acurácia geral de 60% 15 de 25 acertos, os resultados em detalhes podem ser conferidos na Figura 5. O desempenho variou por categoria: 100% em "Investimento – Rendimento do valor", 80% em "Processo – Tramites", 60% em "Atendimento – Qualidade do atendimento"e "Comentários gerais", e 0% em "Processo – Tempo". Trazendo a análise qualitativa nesse cenário, foi identificado que a classe "Processo – Tempo"foi confundida com "Processo – Tempo para Pagamento", sendo classes parececidas, o que reforçou a hipótese da taxonomia. E como uma dupla confirmação os testes com o Qwen foi visto o mesmo padrão.

Classe	Acertos	Acurácia
Atendimento – Qualidade	3/5	60%
Processo – Tramites	4/5	80%
Investimento – Rendimento	5/5	100%
Processo – Tempo	0/5	0%
Comentários Gerais	3/5	60%
Total e media	**15/25**	**60%**

Figura 5: Resultados dos testes realizados com Mistral

Um outro insight identificado foi que em alguns casos o LLM divergiu do "correto", porém ao analisar em detalhes, é identificado que a classificação pelo LLM fez mais sentido, se considerar o padrão do que deve ser preenchido em cada classe, ou seja o que havia sido preenchido pelo analista estava errado e o LLM trouxe o preenchimento correto, isso aconteceu nas classes "Comentários gerais"e no caso divergente de "Processo – Tramites". Isso sugere que os LLMs podem ser ferramentas para auditoria de qualidade de rótulos e para auxiliar no refinamento da taxonomia de categorias.

Então, motivado pelos testes com LLMs e com a confirmação da revisão da taxonomia, foi feita uma revisão manual das Top 10 categorias, as alterações foram: as classes "Comentário genérico"e "Comentários gerais"foram excluídas do dataset, devido a ser uma classe menos importante para a analise de verbatins. - As classes "Processo - Tempo"e "Processo - Tempo para Pagamento", que mostraram confusão nos testes com LLM e a partir da analise da matriz de confusão, foram substituidas pela categoria "Processo - Tramites". - A classe "Produto - Proposta de valor"(minúscula) foi unificada com "Produto - Proposta de Valor"(maiúscula) para corrigir a inconsistência de rotulagem.

Com estas alterações, o dataset original de 10 classes foi transformado em um novo conjunto de dados mais enxuto e semanticamente mais distinto, contendo as Top 5 Categorias e 938 registros.

O modelo Random Forest + TF-IDF foi então retreinado e avaliado neste novo dataset refinado. Os resultados foram comparados diretamente com o desempenho do mesmo modelo no dataset original de Top 10 Categorias. Essas alterações resultaram em uma melhora expressiva em todas as métricas. A Acurácia Geral saltou de 70,67% para 77,66%, a F1-score Macro, subiu de 47,00% para 60,00%, apresentando um aumento expressivo no balanceamento das classes. Este aumento indica que o modelo se tornou mais equilibrado, melhorando seu desempenho médio em todas as classes e não apenas nas majoritárias. Além disso, no novo cenário com 5 classes, nenhuma categoria apresentou recall zero. A matriz de confusão desse cenário pode ser conferido na Figura 6

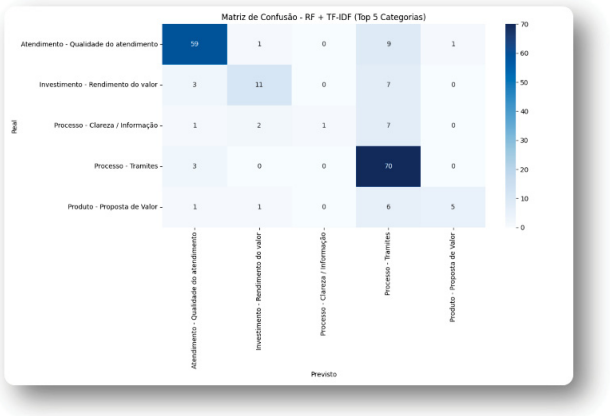

Figura 6: Matriz de confusão do Random Forest com Tf-IDF no cenário de 5 categorias

Um outro ponto identificado nesse teste é que a quantidade de itens por classe está relacionado a taxa de recall, as classes com maior quantidade alcançaram os recalls mais altos. Conforme a quantidade diminui, o recall também tende a cair, a linha de corte para um bom desempenho nesse caso foi 70 itens de teste, conforme a Figura 7

Categoria (Top 5)	Recall (%)	Suporte no Teste (Nº de Linhas)
Processo - Tramites	96%	73
Atendimento - Qualidade do atendimento	84%	70
Investimento - Rendimento do valor	52%	21
Produto - Proposta de Valor	38%	13
Processo - Clareza / Informação	9%	11

Figura 7: Quantidade de Recall nas Top 5 categorias

Os resultados deste experimento validam a hipótese de que a distinção semântica entre as categorias alinhadas a quantidade minima de itens por classe resultam em métricas satisfatorias.

5. Conclusão

Este estudo teve o objetivo de desenvolver e avaliar um sistema de classificação automática para verbatims de clientes do ramo de previdência, comparando modelos de aprendizado de máquina supervisionado

e explorando o potencial de LLMs. A investigação demonstrou que, mesmo que a escolha do algoritmo seja importante, a qualidadedos dados e a clareza da taxonomia das categorias e a consistência da rotulagem humanam, são fatores determinantes para o sucesso da classificação.

A análise comparativa dos modelos supervisionados no cenário focado de 10 categorias revelou que o Random Forest com TF-IDF foi a abordagem mais performática, alcançando a maior acurácia geral (70,67%) e o melhor F1-score Macro (47,00%). Porém ele ainda apresentou dificuldades com classes minoritárias e semanticamente ambíguas. A exploração com o LLM Mistral em modo zero-shot, apesar de uma acurácia de 60% em amostra, foi importante na análise qualitativa para mostrar essas ambiguidades na taxonomia e potenciais inconsistências nos rótulos humanos, servindo como uma ferramenta de diagnóstico.

A principal contribuição deste estudo foi a validação da hipótese de que o refinamento da taxonomia melhora o desempenho do modelo. Ao aplicar as melhorias sugeridas pela análise, o modelo Random Forest + TF-IDF, quando re-treinado no novo dataset de 5 categorias, apresentou uma melhora em todas as métricas, a acurácia geral subiu para 77,66% e o F1-score Macro subiu para 60,00%, não apresentando classes com recall zero neste cenário.

Portanto, o caminho para um sistema de classificação de texto confiável não tem como foco principal a otimização de algoritmos, mas no investimento em uma taxonomia de categorias bem definida e em processos de rotulagem de alta refinados, alinhados assim com a quantidade de dados.

6. Agradecimentos

Meus agradecimentos em primeiro lugar vão a Deus que me possibilitou realizar essa pós graduação, depois a minha esposa que esteve me apoiando e incentivando em todos os momentos, a minha família que também esteve presente em todos os momentos, e por fim ao meu orientador que sempre estava disponível para me ajudar no processo de construção desse estudos.

Referências

[1] Zhang, H. The Optimality of Naive Bayes. In: Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference (FLAIRS), 2004

[2] Hosmer David W., S. Lemeshow, and Rodney X. Sturdivant. Applied Logistic Regression. John Wiley & Sons, 3rd ed., 2013

[3] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2009

[4] Brown et al., Language Models are Few-Shot Learners, 2020