Classificação de Demandas do Fala.BR

Projeto do Bootcamp Machine Learning

Anderson Alves Monteiro - Presidência da República
Léo Maranhão de Mello - SUSEP

Descrição do projeto de machine learning

1- Descrição do problema ou tarefa: O problema consiste em classificar as demandas recebidas pela Ouvidoria da SUSEP, por meio do sistema Fala.BR.

2- Descrição da solução de IA: Utilizamos processamento de NLP e testamos os modelos SGD, XGBoost e Random Forest, todos com treinamento supervisionado, além da utilização e treinamento de uma LLM (BERTimbau). A solução incluiu, dentre outros aspectos, uma pipeline para pré-processamento, treinamento e avaliação do modelo.

3- Fonte de dados: A Ouvidoria forneceu um dataset com 1531 textos das demandas e suas classificações, no formato de planilha ODS. A classificação dos textos foi realizada pela própria Ouvidoria.

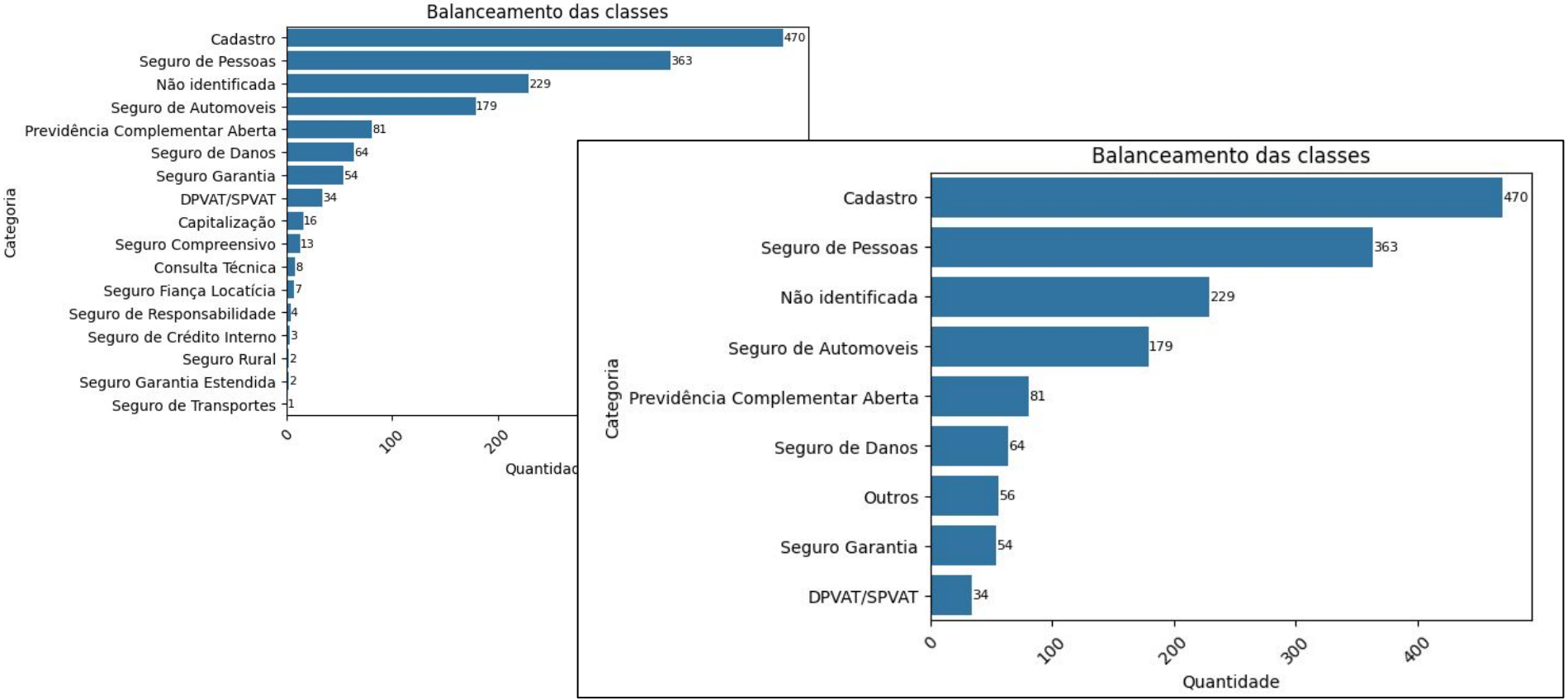
4- Variáveis independentes (preditoras ou "features"): A variável independente é o texto recebido do Fala.BR.

Descrição do projeto de machine learning

5- Variável dependente (resposta ou "target"): A variável dependente é a classificação do texto em uma das 17 classificações utilizadas pela Ouvidoria:

• Cadastro • Capitalização • Consulta Técnica • DPVAT/SPVAT • Não identificada • Previdência Complementar Aberta • Seguro Compreensivo • Seguro de Automóveis • Seguro de Crédito Interno	• Seguro de Danos • Seguro de Pessoas • Seguro de Responsabilidade • Seguro de Transportes • Seguro Fiança Locatícia • Seguro Garantia • Seguro Garantia Estendida • Seguro Rural

Dados (limitações encontradas)



Propostas de solução

1) Classes desbalanceadas + GridSearchCV + SGD/RandomForest/XGBoost + Optuna (f1 - macro)
2) Classes desbalanceadas + GridSearchCV + SGD/RandomForest/XGBoost + Optuna (balanced accuracy)
3) Classes balanceadas com oversampling + GridSearchCV + SGD/RandomForest/XGBoost + Optuna
4) Classes balanceadas com SMOTE + GridSearchCV + SGD/RandomForest/XGBoost + Optuna
5) Classes balanceadas com ChatGPT + GridSearchCV + SGD/RandomForest/XGBoost + Optuna + Ensemble (Llama -> Alpaca)
6) Classes balanceadas com ChatGPT + BERT embeddings + GridSearchCV + SGD/RandomForest/XGBoost + Ensemble
7) Classes balanceadas com ChatGPT + BERT

Pré-processamento

	Proposta			
	ML, Oversampling e SMOTE	**ML, ChatGPT**	**ML, ChatGPT, BERT**	**ChatGPT + BERT**
Pré-processamento	Agrupamento das classes com menor ocorrência; Proposta 3, aplicação do RandomOverSampler; Transformação das classes de texto para número; Separação em teste e treino; Transformação do texto em minúscula, remoção de stop words, remoção de números e sinais de pontuação, lematização e aplicação do TfidfVectorizer; Proposta 4, aplicação do SMOTE.	Agrupamento das classes com menor ocorrência; Transformação das classes de texto para número; Separação em teste e treino; Exposição das classes de treino ao ChatGPT para a geração de novas linhas; Transformação do texto em minúscula, remoção de stop words, remoção de números e sinais de pontuação, lematização e aplicação do TfidfVectorizer.	Agrupamento das classes com menor ocorrência; Transformação das classes de texto para número; Separação em teste e treino; Exposição das classes de treino ao ChatGPT para a geração de novas linhas;	Agrupamento das classes com menor ocorrência; Transformação das classes de texto para número; Separação em teste e treino; Exposição das classes de treino ao ChatGPT para a geração de novas linhas;

Treinamento

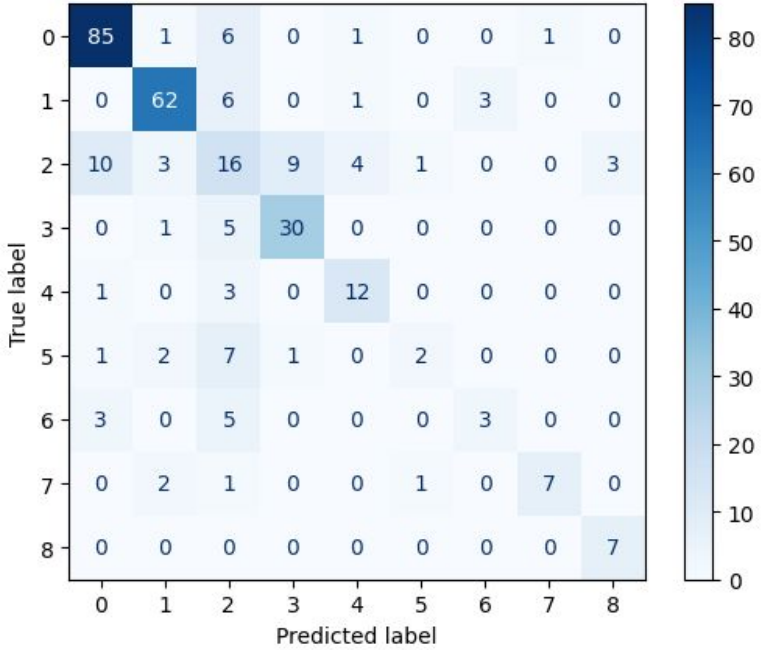
	Proposta			
	ML, Oversampling e SMOTE	**ML, ChatGPT**	**ML, ChatGPT, BERT**	**ChatGPT + BERT**
Treinamento	Após a transformação dos dados de entrada em TFIDF, utilizamos os algoritmos SGDClassifier, RandomForestClassifier e XGBClassifier, inicialmente com o GridSearchCV e depois com o Optuna, para otimizar os hiperparâmetros.	Após a transformação dos dados de entrada em TFIDF, utilizamos os algoritmos SGDClassifier, RandomForestClassifier e XGBClassifier, inicialmente com o GridSearchCV e depois com o Optuna, para otimizar os hiperparâmetros. Ao final fizemos o ensemble do melhor modelo de cada um dos 3 classificadores.	Utilizamos a BERTimbau Base para obter os embeddings dos dados de entrada. Após, utilizamos os algoritmos SGDClassifier, RandomForestClassifier e XGBClassifier com o GridSearchCV. Ao final fizemos o ensemble do melhor modelo de cada um dos 3 classificadores.	Utilizamos os dados de entrada para retreinar a última camada da BERTimbau Base para as categorias dos nossos dados.

Resultado

	Proposta			
	ML, Oversampling e SMOTE	**ML, ChatGPT**	**ML, ChatGPT, BERT**	**ChatGPT + BERT**
Resultado	As propostas 1, 2, 3 e 4 obtiveram um resultado inferior. Apresentamos os resultados do melhor classificador entre as 4 propostas: Proposta 4: XGBClassifier(random_state=42, eval_metric='logloss'), hiperparâmetros: {'classifier__learning_rate': 0.1, 'classifier__max_depth': 4, 'classifier__n_estimators': 400} F1 - macro: 0,64	Melhor resultado entre todas as propostas. Melhor classificador: SGDClassifier(alpha=1e-06, random_state=42), hiperparâmetros: {'clf__loss': 'log_loss', 'clf__penalty': 'l2'} F1 - macro: 0,91	Melhor classificador foi o ensemble dos melhores modelos ajustados: VotingClassifier, voting='hard': SGDClassifier(alpha=1e-06, random_state=42), hiperparâmetros: {'clf__loss': 'hinge', 'clf__penalty': 'elasticnet'} RandomForestClassifier(random_state=42), hiperparâmetros: {'classifier__max_depth': 20, 'classifier__max_features': 'sqrt', 'classifier__min_samples_leaf': 2, 'classifier__min_samples_split': 2, 'classifier__n_estimators': 100} XGBClassifier(random_state=42, eval_metric='logloss'), hiperparâmetros: {'classifier__learning_rate': 0.1, 'classifier__max_depth': 4, 'classifier__n_estimators': 350} F1 - macro: 0,91	A LLM foi treinada com os seguintes argumentos: num_train_epochs=3 per_device_train_batch_size=16 per_device_eval_batch_size=64 warmup_steps=500 weight_decay=0.01 logging_steps=10 evaluation_strategy="epoch" F1 - macro: 0,81

Avaliação - Propostas 1, 2, 3 e 4 - ML, Oversampling e SMOTE

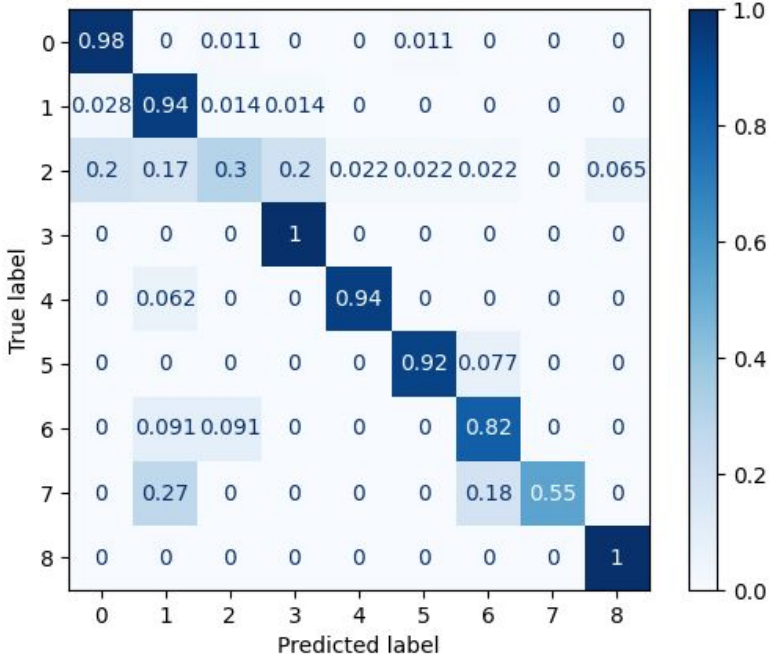
	precision	recall	f1-score	support
Cadastro	0.85	0.90	0.88	94
Seguro de Pessoas	0.87	0.86	0.87	72
Não identificada	0.33	0.35	0.34	46
Seguro de Automoveis	0.75	0.83	0.79	36
Previdência Complementar Aberta	0.67	0.75	0.71	16
Seguro de Danos	0.50	0.15	0.24	13
Outros	0.50	0.27	0.35	11
Seguro Garantia	0.88	0.64	0.74	11
DPVAT/SPVAT	0.70	1.00	0.82	7
accuracy			0.73	306
macro avg	0.67	0.64	0.64	306
weighted avg	0.73	0.73	0.72	306



Avaliação - Proposta 7 - ChatGPT + BERT

Relatorio de Classificacao:

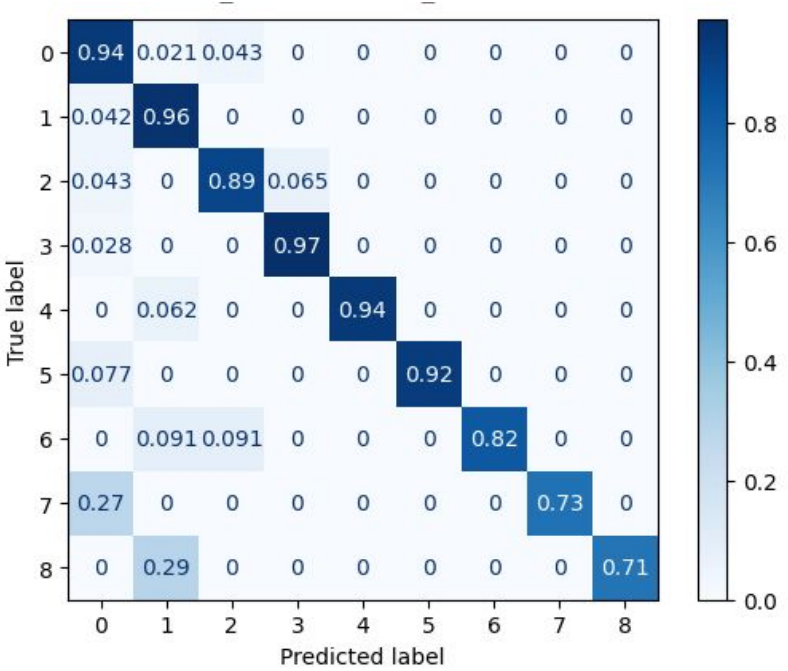
	precision	recall	f1-score	support
Cadastro	0.89	0.98	0.93	94
Seguro de Pessoas	0.84	0.94	0.89	72
Não identificada	0.82	0.30	0.44	46
Seguro de Automoveis	0.78	1.00	0.88	36
Previdência Complementar Aberta	0.94	0.94	0.94	16
Seguro de Danos	0.86	0.92	0.89	13
Outros	0.69	0.82	0.75	11
Seguro Garantia	1.00	0.55	0.71	11
DPVAT/SPVAT	0.70	1.00	0.82	7
accuracy			0.85	306
macro avg	0.84	0.83	0.81	306
weighted avg	0.85	0.85	0.82	306



Avaliação - Proposta 6 - ML, ChatGPT, BERT

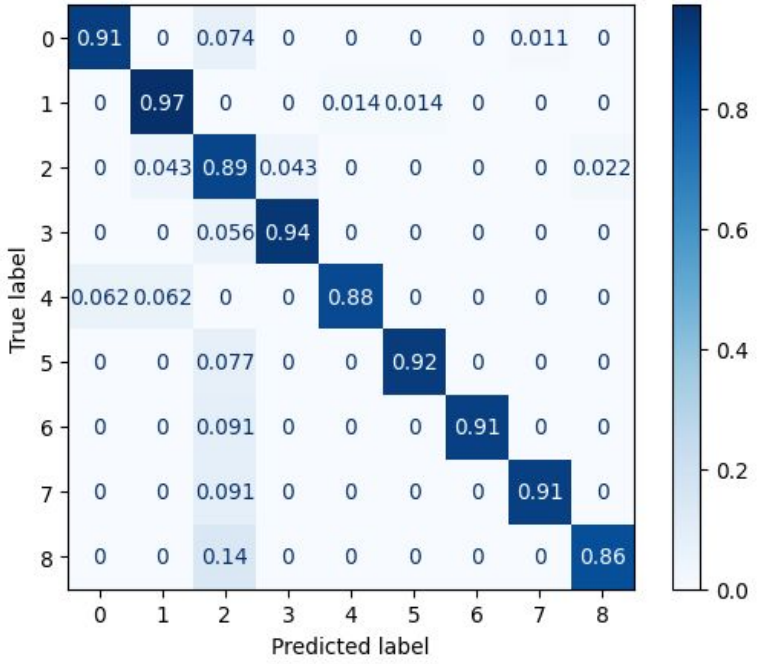
Relatório de Classificação - Ensemble (Voting Classifier):

	precision	recall	f1-score	support
Cadastro	0.90	0.94	0.92	94
Seguro de Pessoas	0.92	0.96	0.94	72
Não identificada	0.89	0.89	0.89	46
Seguro de Automoveis	0.92	0.97	0.95	36
Previdência Complementar Aberta	1.00	0.94	0.97	16
Seguro de Danos	1.00	0.92	0.96	13
Outros	1.00	0.82	0.90	11
Seguro Garantia	1.00	0.73	0.84	11
DPVAT/SPVAT	1.00	0.71	0.83	7
accuracy			0.92	306
macro avg	0.96	0.88	0.91	306
weighted avg	0.92	0.92	0.92	306



Avaliação - Proposta 5 - ML, ChatGPT

	precision	recall	f1-score	support
Cadastro	0.99	0.91	0.95	94
Seguro de Pessoas	0.96	0.97	0.97	72
Não identificada	0.76	0.89	0.82	46
Seguro de Automoveis	0.94	0.94	0.94	36
Previdência Complementar Aberta	0.93	0.88	0.90	16
Seguro de Danos	0.92	0.92	0.92	13
Outros	1.00	0.91	0.95	11
Seguro Garantia	0.91	0.91	0.91	11
DPVAT/SPVAT	0.86	0.86	0.86	7
accuracy			0.92	306
macro avg	0.92	0.91	0.91	306
weighted avg	0.93	0.92	0.93	306



Interpretação

As propostas 5 e 6 apresentaram resultados semelhantes, sendo a escolha pela classe 5 por conta da matriz de confusão, que apresentou um resultado melhor nas categorias “DPVAT/SPVAT” e “Seguro Garantia”.

O melhor modelo da proposta 5 classificou, considerando a média ponderada das classes, 93% das vezes sem gerar falso positivo, e 92% das vezes na categoria correta.

A proposta 5 apresentou 23 classificações incorretas, sendo que 18 envolvem a categoria 2 - “Não Identificadas”. Analisando a classificação realizada pela Ouvidoria, parece que critérios variados foram utilizados durante a classificação manual. Tais critérios podem ser melhorados e trariam reflexo em uma classificação mais adequada pelos algoritmos.

Aspectos interessantes, benefícios dos resultados, insight obtido, próximos passos, onde será publicado.

O aspecto interessante foi o desenvolvimento dos prompts para a geração de dados pelo GhatGPT. Dependendo da categoria usada para a classificação, o nível de RAG precisou ser mais detalhado.

Vimos a importância de uma quantidade maior de dados para o desempenho dos modelos de classificação.

O trabalho realizado ajudará a Ouvidoria da SUSEP, que hoje mantém um grupo de pessoas para a realização da classificação manual, faltando estabelecer o processo de trabalho no qual o algoritmo de ML será incluído.