

# Classificação Automática de Petições Iniciais Usando Classificadores Combinados

Matheus Serrão Marinato<sup>1</sup>, Antonio F. L. Jacob Junior<sup>1</sup>  
Fábio M. F. Lobato<sup>1,2</sup>, Omar A. C. Cortes<sup>1,3</sup>

<sup>1</sup>Centro de Ciências Tecnológicas – Universidade Estadual do Maranhão (UEMA)  
São Luis, MA – Brasil

<sup>2</sup>Instituto de Engenharia e Geociências – Universidade Federal do Oeste do Pará (UFOPA)  
Santarém, PA – Brasil

<sup>3</sup>Departamento de Computação – Instituto Federal do Maranhão (IFMA)  
São Luis, MA – Brasil

antoniojunior@professor.uema.br, omar@ifma.edu.br

**Resumo.** *É expressivo o aumento da demanda judicial e da escassez de recursos, prejudicando o atendimento e a agilidade do sistema jurídico brasileiro. Dessa forma, é evidente a necessidade de se investir em novas aplicações tecnológicas para a garantia do bom andamento processual. Assim, a correta classificação das petições facilita a categorização do processo em sua fase inicial, evitando futuras demandas repetitivas que trataram sobre a mesma situação. Nesse contexto, esta pesquisa tem o objetivo de utilizar aprendizagem de máquina combinada de modo a classificar petições iniciais, dando celeridade ao fluxo processual. Para tal, uma base de dados com 1.787 modelos de petições foi construída a partir de diversos portais com conteúdo jurídico. Os resultados mostram que o classificador combinado apresentou o melhor desempenho, sendo 90% de acurácia, 91% de precisão, 87% de recall e 89% de F1-Score. Conclui-se que o modelo tem o potencial de ser utilizado para prover maior celeridade na distribuição processual.*

## 1. Introdução

No relatório Justiça em Números, percebe-se a criticidade do aumento da demanda judicial e escassez de recursos, pois o número de processos pendentes na Justiça vem aumentando a cada ano [CNJ 2021]. Isto revela problemas de superlotação de processos nas unidades judiciárias, o que impacta na eficiência do atendimento dos anseios e direitos da sociedade [Hollanda and Leite 2020]. Em termos quantitativos, existem cerca de 74,5 milhões de processos em tramitação, com uma média de 27,9 milhões de novos casos, não tendo como reduzir significativamente a quantidade dos processos em tramitação em um curto espaço de tempo sem o auxílio de novas tecnologias [CNJ 2021].

Os processos são compostos pela petição inicial, que é o instrumento pelo qual o indivíduo convoca o poder Judiciário para atuar em um caso concreto. As petições precisam ser redigidas de forma clara e objetiva, isto é, de forma concisa, empregando a linguagem correta, utilizando a melhor terminologia jurídica e tendo como preocupação o sentido da construção, estando apta a expor o seu conteúdo e raciocínio jurídico [Campestrini and Florence 2000]. Nela, o profissional do direito deve sempre ficar atento

às regras impostas no código processual vigente, sempre tendo a preocupação de consultar a legislação, a doutrina e citações de jurisprudências [Campestrini and Florence 2000].

Neste contexto, a classificação da petição inicial ocorre de acordo com as características da sua natureza, sendo realizada no ato do cadastro do documento digital. Como etapa inicial são analisados o objeto e os pedidos contidos na petição. O primeiro assunto informado deve ser aquele correspondente ao tema principal que dará característica ao processo [Palaia 2000]. Esta etapa é propensa à erros, devido sua subjetividade e por ser realizada de forma manual. Por isso, a correta classificação dos assuntos das petições pode auxiliar na busca por petições similares e suas decisões, reduzindo o tempo que é levado para encontrar uma informação nos sistemas da justiça.

Outra tarefa que pode ser facilitada pelo processo de classificação dos assuntos é a triagem inicial [Leme 2021]. Neste momento, a petição chega ao gabinete e precisa ser direcionada para o juiz que trabalha naquele tipo de causa. Se ao chegar no gabinete o assunto já estiver automaticamente categorizado, o seu direcionamento fica facilitado, pois poupa a leitura completa da petição inicial ou do recurso pelo servidor responsável pela triagem, possibilitando até mesmo a criação de mecanismos automáticos para a distribuição dos processos [Faraco 2020].

Dessa forma, para auxiliar na melhoria da classificação, alguns estudos envolvendo aprendizagem de máquina vêm sendo realizados no âmbito jurídico. Por exemplo, em [Almeida Neto et al. 2018] classificam-se processos da justiça estadual como procedentes ou improcedentes. Em [Leme 2021] classifica-se o rito do processo, podendo ser sumário ou ordinário. Em [Sousa 2019] faz-se a classificação das petições iniciais usando quatro algoritmos tradicionais de aprendizagem de máquina (Árvore de Decisão, Naïve-Bayes, *Support Vector Machine* (SVM) e *k-Nearest Neighbors* (KNN)). Em termos de implementação efetiva em ambiente de produção, pode-se citar o Tribunal de Justiça do Amazonas que realiza a classificação de petições intermediárias [Justiça Digital 2021].

Como se pode observar, aplicações de aprendizagem de máquina vêm sendo aplicadas nos tribunais de justiça com o objetivo de acelerar a tramitação dos processos. No entanto, os trabalhos citados utilizam algoritmos tradicionais para realizar a tarefa de classificação. Com isso, percebeu-se uma lacuna na literatura quanto ao uso de modelos combinados, uma vez que este tipo de técnica proporciona um melhor desempenho na classificação. Nesse cenário, este trabalho propõem a utilização de classificadores combinados de modo a obter resultados mais precisos na classificação de petições iniciais.

O restante do artigo encontra-se organizado como segue. A Seção 2 mostra a teoria geral dos classificadores combinados. A Seção 3 apresenta a proposta de um classificador combinado para classificação de petições iniciais e o *framework* experimental. Os resultados são apresentados e discutidos na Seção 4. Finalmente, a Seção 5 apresenta as conclusões do trabalho e sugestões de pesquisas futuras.

## 2. Classificadores Combinados

Classificadores combinados, do inglês *ensemble classifiers*, é uma técnica de aprendizado de máquina que tenta produzir um modelo preditivo ideal, tendo como objetivo melhorar a precisão dos resultados por meio da combinação vários modelos [Zhou 2009]. No pior caso, o classificador combinado tende a apresentar resultados semelhantes aos modelos

tradicionais. Existem três métodos que dominam o campo de estudo dos classificadores combinados: *bagging*, *boosting* e *stacking*.

O *Bagging* é um tipo de técnica em que um único algoritmo de treinamento é usado em diferentes subconjuntos de dados, sendo que a amostragem do subconjunto dados é feita com substituição. Uma vez que o algoritmo é treinado em todos os subconjuntos, o algoritmo faz a predição agregando todas as saídas geradas pelos diferentes modelos, sendo mais comum a votação em tarefas de classificação e a média das saídas em tarefas de regressão. O algoritmo mais conhecido desse tipo é o *Random Forest*, ou Floresta Aleatória [Breiman 2001].

Já o *Boosting* é um método de conjunto para melhorar as previsões do modelo de qualquer algoritmo de aprendizado. A ideia é treinar classificadores fracos sequencialmente, cada um tentando corrigir seu predecessor, fazendo com que os classificadores fracos sejam convertidos em aprendizes fortes [Zhou 2009]. Nesse tipo de classificador, um algoritmo que tem apresentado bons resultados e se tornado popular é o XGBoost [Chen and Guestrin 2016].

No *stacking*, utilizam-se dois ou mais modelos de base, geralmente chamados de modelos de nível 0, que se ajustam aos dados de treinamento e cujas previsões são compiladas. Um metamodelo que combina as previsões dos modelos de base, chamados de modelo de nível 1, aprende a melhor combinação dos modelos base. Como a possibilidade de combinação de algoritmos e modelos é vasta, é difícil de sugerir um único algoritmo como nos outros métodos, no entanto, trabalhos como o de [Liang et al. 2021] e [Yi et al. 2020] tendem a mostrar que a abordagem *stacking* é mais estável que as demais.

### **3. Classificador combinado para classificação de petições iniciais**

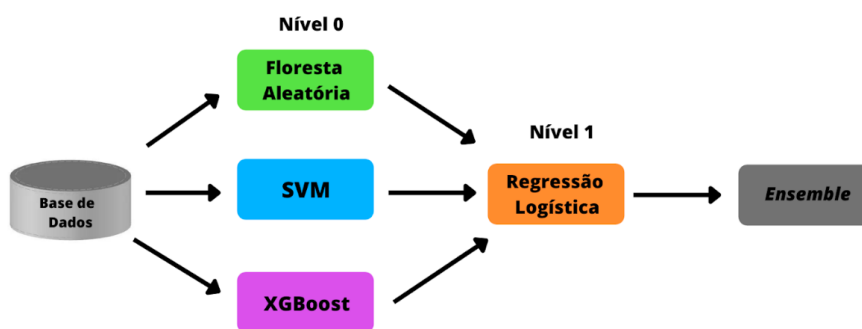
Nesta seção são descritos o classificador combinado do tipo *stacking* para realizar a classificação de petições iniciais utilizado no presente estudo e o *framework* experimental adotado. Na implementação e execução dos testes utilizaram-se o ambiente *Google Colab* com a linguagem Python e as bibliotecas *scikit-learn* de aprendizagem de máquina e *Natural Language Toolkit* (NLTK) para o processamento de linguagem natural.

#### **3.1. Abordagem proposta**

Nesse contexto, a Figura 1 apresenta um diagrama do modelo proposto, na qual estão dispostos os níveis do classificador e o metamodelo combinador. O nível 0 é composto por três algoritmos, sendo que a escolha deles se deu por representatividade. A Floresta Aleatória é um algoritmo do tipo *bagging*, o XGBoost um do tipo *boosting* e o SVM representa um algoritmo tradicional bem conhecido na literatura de aprendizagem de máquina. Por este motivo, a presente proposta consiste em um *ensemble* de *ensemble*. No nível 1 a regressão logística é utilizada para aprender a melhor forma de combinar os resultados. Para que os dados pudessem ser manipulados pelo nível 0, os textos das petições iniciais passaram por um processo de preparação, o qual é apresentado com detalhes a seguir.

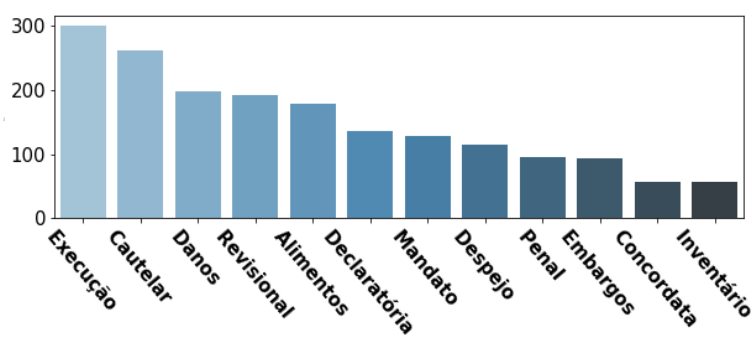
#### **3.2. Base de dados e Pré-processamento**

Uma base de dados com 1.787 modelos de petições foi construída a partir de diversos portais com conteúdo jurídico. Cada tipo de petição era composto por modelos distintos



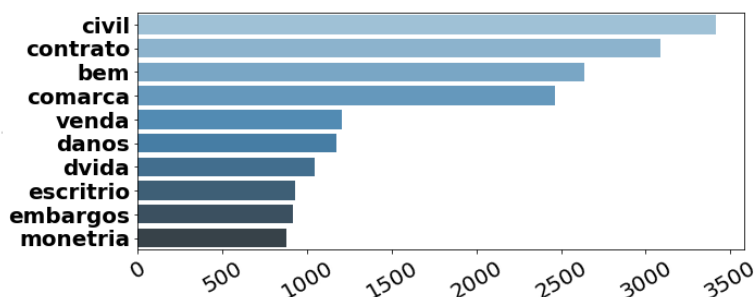
**Figura 1. Classificador proposto do tipo *Stacking*.**

no formato “pdf”, sendo compostas por: 1) cabeçalho - que trata da informação da natureza da petição; 2) corpo da petição - em que é exposto quais os pedidos solicitados pelo requerente. Foram coletados vários tipos de petições conforme mostra a Figura 2, na qual observa-se que os tipos mais comuns de petição são as de “*execução*” e a de “*medida cautelar*”. Esta concentração torna a aprendizagem particularmente desafiador.



**Figura 2. Distribuição dos tipos de petições na base de dados construída.**

É importante destacar que para cada tipo de petição existem vários termos que podem ser utilizados para classificá-las. A Figura 3 ilustra as principais palavras que aparecem nas petições. Uma das palavras mais comuns é “*bem*”, que é comum em petições do tipo inventário. Como a variação de palavras é grande, isso também representará um desafio para o algoritmo em criar um modelo com bons resultados, por isso a importância de realizar uma avaliação criteriosa do pré-processamento dos dados.



**Figura 3. Palavras mais frequentes no *corpus*.**

Na etapa de pré-processamento, o primeiro passo na preparação foi converter os

arquivos “pdf” em “txt”. Após a leitura dos arquivos, foi criada uma tabela contendo dois atributos: 1) tipo - contendo o assunto da petição; e 2) petição - armazenando os demais dados da petição. Com o *corpus* tabulado, passou-se então para a limpeza de dados, a qual foi baseada em [Cirqueira et al. 2018]. No presente estudo retiraram-se os espaços em branco, pontuações, símbolos financeiros, caracteres especiais e as *stop words*. A retirada das *stop words* é uma etapa importante, visto que reduz o espaço de busca e melhora significativamente o processo de aprendizagem do algoritmo [Sousa et al. 2019], e, até mesmo, a explicabilidade [Cirqueira et al. 2020]. Para representação dos dados optou-se pelo *Term Frequency - Inverse Document Frequency* (TF-IDF) dada a sua simplicidade e vasta aplicabilidade como em [Sousa et al. 2019, Cirqueira et al. 2020].

### 3.3. Parametrização

Os experimentos foram conduzidos utilizando validação cruzada (*cross-validation*) com método de amostragem *k-fold* com  $k = 10$ . Os hiper-parâmetros de cada algoritmo foram empiricamente escolhidos e são apresentados na Tabela 1.

**Tabela 1. Hiper-parâmetros dos algoritmos de aprendizagem.**

Algoritmo	Parametrização
Regressão Logística	<i>class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, n_jobs=None</i>
Floresta Aleatória	<i>random_state=42, n_jobs=-1, max_depth=100, n_estimators=1000, oob_score=True</i>
SVM	<i>C=1.0, kernel='rbf', degree=30, coef0=0.0, shrinking=True, probability=False, tol=0.01, cache_size=200, class_weight=None, verbose=False, max_iter=-1, random_state=None</i>
XGBoost	<i>colsample_bytree=0.7, learning_rate=0.1, alpha=10, max_depth=20, n_estimators=300</i>

## 4. Resultados

Para a avaliação dos modelos, o presente trabalho utilizou métricas bem consolidadas na avaliação de classificadores, a saber: acurácia, precisão, *recall* e F1-score. A Tabela 2 mostra um comparativo do classificador combinado com cada um dos algoritmos usados de forma independente. Os melhores resultados encontram-se destacados em negrito.

**Tabela 2. Resultados para as medidas de desempenho adotadas.**

Algoritmo	Acurácia	Precisão	Recall	F1-Score
Regressão Logística	83%	89%	74%	78%
Floresta Aleatória	86%	90%	82%	84%
SVM	88%	86%	74%	77%
XGBoost	83%	81%	76%	77%
Classificador Combinado	<b>90%</b>	<b>91%</b>	<b>87%</b>	<b>89%</b>

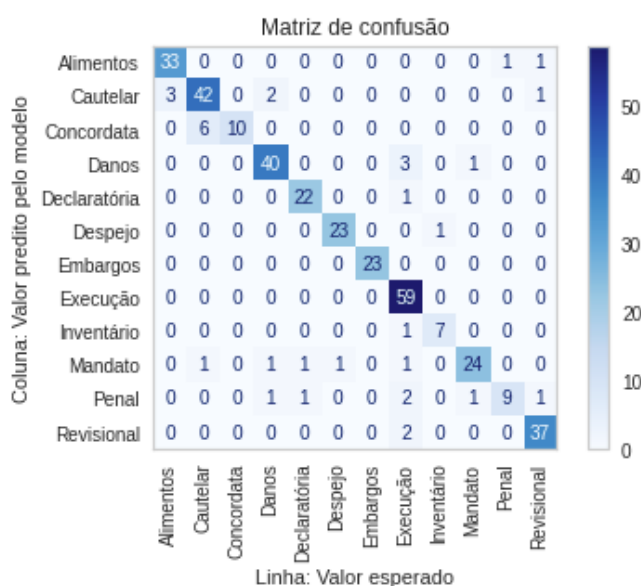
Como pode ser observado, o classificador combinado alcança as melhores métricas, sendo 90%, 91%, 87% e 89% na acurácia, precisão, *recall* e F1-Score, respectivamente. A segunda melhor acurácia é do SVM; a segunda melhor precisão, *recall* e F1-Score é da Floresta Aleatória. Neste cenário, o modelo proposto (*ensemble* de *ensemble*) consegue combinar o melhor dos algoritmos e fornecer o melhores resultados no domínio de aplicação estudado.

Considerando que o classificador combinado obteve o melhor resultado, a Tabela 3 apresenta o seu desempenho com relação a cada tipo de petição. Os resultados não refletem a quantidade de tipos petições. Por exemplo, petições dos tipos “*execução*” e “*cautelar*” são as que apresentam maior quantidade de amostras na base de dados, porém, não são as que apresentam maior F1-Score. Um fato que possivelmente influenciou no desempenho geral é o tipo de palavras mais frequentes, por exemplo, “*bem*” que é uma palavra frequente nas petições de “*inventário*”, o qual teve precisão de 100%.

**Tabela 3. Desempenho do classificador combinado por tipo de petição.**

Tipos	Precisão	Recall	F1-Score
Alimentos	97%	94%	96%
Cautelar	88%	96%	91%
Concordata	87%	80%	93%
Danos	95%	85%	90%
Declaratória	79%	92%	85%
Despejo	96%	<b>100%</b>	98%
Embargos	83%	<b>100%</b>	91%
Execução	75%	93%	84%
Inventario	<b>100%</b>	88%	93%
Mandato	<b>100%</b>	90%	95%
Penal	<b>100%</b>	75%	86%
Revisional	93%	85%	<b>99%</b>

Por fim, a Figura 4 apresenta a matriz de confusão por tipo de petição, na qual pode-se observar que as do tipo “*penal*” foram as mais difíceis de serem classificadas, impactando negativamente no desempenho do modelo, fato que, também, pode ser observado na métrica *recall*. Por outro lado, as petições do tipo “*embargo*” e “*execução*” foram as que obtiveram a melhor classificação, sendo esta última o tipo com maior quantidade de petições na base de dados. Outrossim, “*concordata*” e “*inventario*” são as que apresentam a menor quantidade de exemplos dos tipo estudados.



**Figura 4. Matriz de confusão por tipo de petição.**

## 5. Conclusões

Este trabalho apresentou o desenvolvimento de uma abordagem do tipo *stacking* para melhorar a classificação de petições iniciais dos tribunais de justiça. Os resultados mostraram que o classificador combinado alcançou melhores resultados com uma acurácia de 90%, precisão de 91%, 87% de *recall* e 89% de *F1-Score*.

Conclui-se então que o modelo proposto tem o potencial de ser utilizado para prover maior celeridade na distribuição processual. Nesse contexto, este classificador pode ser adotado como ferramenta a ser implantada por Tribunais de Justiça, visto que apresentou resultados promissores e há escassez de ferramentas capazes de auxiliar nessa tarefa. O uso do modelo poderia prover melhor acuidade e celeridade na distribuição das demandas e tomada de decisão sentencial.

Como trabalhos futuros, pretende-se melhorar o desempenho da aplicação por meio da inclusão de mais classificadores na abordagem *stacking*; pode-se, também, utilizar algoritmos de otimização como o *Tabu Search*, para selecionar quais classificadores utilizar no modelo *ensemble*. O *Tabu Search* é particularmente interessante, visto que constrói uma lista de parâmetros já utilizados, evitando assim que os mesmos classificadores sejam escolhidos repetidamente. Além disso, será realizada a apresentação dos resultados para o Tribunal de Justiça do Maranhão, no qual alguns dos autores possuem acordo de cooperação, a fim de se avaliar a viabilidade de usar o modelo proposto em suas práticas laborais.

## Agradecimentos

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) -DT-308334/2020; pela Fundação Amazônia de Amparo a Estudos e Pesquisas (FAPESPA) PRONEM-FAPESPA/CNPq nº 045/2021; e Acordo de Cooperação Técnica Nº 02/2021 (Processo Nº 38328/2020 -TJ/MA). Agradecemos também aos revisores(as) pelas sugestões que muito auxiliaram na melhora do trabalho.

## Referências

- Almeida Neto, M., Moura, V., Bandeira, J., Freitas, P., and Fagundes, R. (2018). Um modelo de inferência para a classificação de resultados processuais da justiça estadual. *Macabéa-Revista Eletrônica do Netlli*, 3(3).
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- Campestrini, H. and Florence, R. C. (2000). *Como redigir petição inicial*. Saraiva Educação SA, São Paulo, SP.
- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Cirqueira, D., Almeida, F., Cakir, G., Jacob, A., Lobato, F., Bezbradica, M., and Helfert, M. (2020). Explainable sentiment analysis application for social media crisis management in retail. In *4th International Conference on Computer-Human Interaction Research and Applications - Volume 1: WUDESHI-DR*.

- Cirqueira, D., Fontes Pinheiro, M., Jacob, A., Lobato, F., and Santana, Á. (2018). A literature review in preprocessing for sentiment analysis for brazilian portuguese social media. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*.
- CNJ (2021). *Conselho Nacional de Justiça. Justiça em números 2021*. CNJ, Brasília.
- Faraco, F. M. (2020). Modelo de conhecimento baseado em tópicos de acórdãos para suporte à análise de petições iniciais. Master's thesis, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil.
- Hollanda, Y. R. d. and Leite, F. G. d. F. (2020). Petição Inicial: uma análise à luz de teorias bakhtinianas. *Macabéa-Revista Eletrônica do Netlli*, 9(4):292–308.
- Justiça Digital (2021). Tjam automatiza classificação de petições intermediárias no portal e-saj, <https://justicadigital.com/tjam-ia-peticionamento/>, visitado em 28/11/2021.
- Leme, B. (2021). Classificação automática de documentos de características econômicas para defesa jurídica. Master's thesis, Universidade de São Paulo, São Paulo, SP, Brasil.
- Liang, M., Chang, T., An, B., Duan, X., Du, L., Wang, X., Miao, J., Xu, L., Gao, X., Zhang, L., et al. (2021). A stacking ensemble learning framework for genomic prediction. *Frontiers in genetics*, 12.
- Palaiá, N. (2000). *Técnica da petição inicial*. Saraiva Educação SA, São Paulo, SP.
- Sousa, G. N. d., Almeida, G. R., and Lobato, F. (2019). Social network advertising classification based on content categories. In *International Conference on Business Information Systems*, pages 396–404. Springer.
- Sousa, R. N. d. (2019). Minerjus: solução de apoio à classificação processual com uso de inteligência artificial. Master's thesis, Universidade Federal do Tocantins, Palmas, TO, Brasil.
- Yi, H.-C., You, Z., Wang, M.-N., Guo, Z.-H., Wang, Y., and Zhou, J.-R. (2020). Rpi-se: a stacking ensemble learning framework for ncna-protein interactions prediction using sequence information. *BMC Bioinformatics*, 21.
- Zhou, Z. H. (2009). *Ensemble Learning, Encyclopedia of Biometrics*, volume 1. Springer, Berlin, Germany.