Prevendo Decisões Judiciais em Tribunais Brasileiros de Primeira Instância com Aprendizagem de Máquina

Marcelo Gomes Pereira de Lacerda [1] orcid.org/0000-0002-6087-2770

Camila Barros Couceiro d'Amorim [2] orcid.org/0000-0002-4270-0625

Arthur Felipe Melo Alvim [3] orcid.org/0000-0002-9836-7243

[1] Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil,
[2] Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil,
[3] Intelivix LTDA, Recife, Brasil.

E-mail do autor principal: Marcelo Lacerda mgpl@cin.ufpe

Resumo

Um dos maiores problemas enfrentados por escritórios de advocacia ou setores jurídicos de empresas é o planejamento do contingenciamento de recursos, dadas as perspectivas de sentenciamento dos seus processos jurídicos. Atualmente, um grupo de advogados, através de processos completamente subjetivos e não-padronizados, analisa um grande volume de processos e emitem um parecer sobre o que pode vir a ocorrer nos próximos meses quanto às possíveis condenações. Portanto, decisões sobre contingenciamento são tomadas em cima de processos de inferência pouco rigorosos. Este artigo apresenta modelos de previsão de sentenças jurídicas de tribunais brasileiros de primeira instância construídos a partir de 61 bases de dados provenientes de todas as regiões do Brasil, totalizando mais de 600 mil processos. Para a construção destas bases, foram utilizados apenas dados disponíveis publicamente nos sites dos tribunais. Para a construção destes modelos, foram utilizados *ensembles* de Árvores de Decisão, cujos hiperparâmetros foram ajustados pelo método *Population Based Training*. Os resultados mostraram que existe uma grande variabilidade na complexidade das bases espalhadas pelo país, o que gera modelos com qualidades completamente diferentes entre si. Neste estudo, nossos modelos apresentaram precisões que variaram de 36% a 79%.

Palavras-Chave: Previsão de decisões jurídicas; Aprendizagem de Máquina; Árvores de Decisão.

Abstract

One of the biggest problems faced by law firms or legal departments is the planning of resource contingency, given the prospects of their legal processes. Currently, a group of lawyers, through a completely subjective and non-standardized processes, analyze a large number of cases and issue an opinion on what may happen in the coming months regarding possible sentences. Therefore, contingency decisions are made based on non-rigorous inference processes. This article presents an extensive set of experiments on prediction of legal judgments in brazilian courts of first instance using models built from 61 databases from all regions of Brazil, totaling more than 600 thousand cases. This database was built with only publicly available data at the courts' websites. For the construction of these models, ensembles of Decision Trees were used, whose hyperparameters were adjusted by the Population Based Training method. The results showed that the complexity of the problem varies significantly across the country, depending on the available data. Such a difference causes the trained models to show different accuracies. In this study, the models' accuracies varied from 36% to 79%.

Key-words: *Prediction of legal cases outcomes; Machine learning; Decision trees.*

DOI: 10.25286/repa.v6i3.1725

1. Introdução

A Inteligência Artificial vem contribuindo crescentemente para a automação de atividades cada vez menos mecânicas e cada vez mais dependentes de um maior refinamento cognitivo, antes exclusivo dos humanos. Na área do Direito, robôs vêm ajudando advogados e clientes em suas atividades executadas durante o decorrer de um processo jurídico. Entre diversas possíveis aplicações dentro desta área, a previsão do resultado de processos está entre os problemas que recebem mais atenção da indústria e da academia.

Com este auxílio, empresas podem traçar um plano mais preciso de contingenciamento de recursos financeiros para poder arcar com processos que provavelmente, ou até possivelmente, resultarão em condenação com pagamentos de indenização, multas, etc. Atualmente, em muitas empresas, os cálculos de contingenciamento são feitos a partir de análises não padronizadas e superficiais realizadas por advogados, resultando em um contingenciamento consideravelmente menor ou maior do que o necessário. Além disso, empresas podem, através de modelos de previsão bem construídos, fazer simulações em cenários hipotéticos e, através de algoritmos de busca e otimização, traçar as melhores estratégias para cada caso em particular.

O problema de previsão de sentença jurídica vem recebendo cada vez mais atenção na indústria e na Academia. Aos poucos, a comunidade científica vem compartilhando bases de dados para servirem de *benchmark* em experimentos que validam novas técnicas de previsão de sentenças. Como bases recorrentemente utilizadas, podemos destacar uma base de processos da Corte Europeia de Direitos Humanos (ECHR) **[1][2][3][4][5][6][7]**, e um conjunto de bases de processos criminais da Suprema Corte chinesa **[8][9][10][11]**.

Quanto à abordagem utilizada para a construção dos modelos de previsão, a maior parte das publicações até o momento fazem uso de técnicas de aprendizagem de máquina mais tradicionais, como *Support Vector Machines* (SVM) **[1][2][5][6][8][12][13]**, *Naïve Bayes* **[3][6][12][14]**, *Multi-layer Perceptron* (MLP) **[6]**, *k-Nearest Neighbors* (kNN) **[6]** e Árvores de Decisão **[6][1]**. Além disso, pode ser observado na literatura uma recorrência na aplicação de Florestas de Árvores de Decisão **[6][7][13][15]** e métodos de Aprendizagem Profunda **[4][5][8][16]** no problema em questão.

Quanto aos resultados alcançados, os mesmos variam fortemente de acordo as características da base. Nos artigos consultados e citados neste trabalho, foram encontrados modelos com acurácias variando entre 50% e 98%. Porém, em muitos dos trabalhos que apresentaram acurácias maiores de 80%, as F1 *scores* apresentaram baixos valores, o que revela um desequilíbrio entre precisão e *recall* dos modelos preditivos. De qualquer forma, esta grande variância revela que não se pode definir um patamar único que representa o estado-da-arte em previsão de sentenças jurídicas, pois as bases possuem complexidades totalmente diferentes. Portanto, algoritmos só devem ser comparados entre si se aplicados às mesmas bases.

Em **[17]**, tem-se o único artigo publicado encontrado onde são construídos modelos de previsão de sentenças jurídicas em tribunais brasileiros, o qual foi publicado em 2019. Porém, o mesmo apresenta resultados de previsões apenas para o Supremo Tribunal Federal. Nenhum trabalho com modelos de previsão de processos de primeira instância em tribunais brasileiros foi jamais publicado.

Este artigo apresenta modelos de previsão de sentenças jurídicas de tribunais brasileiros de primeira instância construídos a partir de 61 bases de dados provenientes de todas as regiões do Brasil, totalizando mais de 600 mil processos. Para a construção destas bases, foram utilizados apenas dados disponíveis publicamente nos sites dos tribunais. Para a construção destes modelos, foram utilizados *ensembles* de Árvores de Decisão, cujos hiperparâmetros foram ajustados pelo método *Population Based Training*.

Este artigo está estruturado da seguinte forma: na seção 2, encontra-se a descrição da metodologia utilizada durante todo o processo de construção dos modelos, além do detalhamento das bases de dados utilizadas; na seção 3, encontram-se os resultados e as discussões dos mesmos; por fim, na seção 4, as conclusões deste trabalho são apresentadas.

2. Metodologia

Esta seção descreve a metodologia utilizada para a criação dos modelos de previsão, a qual está dividida em 3 subseções: Base de Dados, a qual apresenta as bases de dados utilizadas nos experimentos; Pré-

Processamento e *Feature Engineering*, que explana as etapas anteriores à modelagem; e Modelagem, que apresenta o método utilizado para modelagem e seleção de hiperparâmetros e modelos gerados.

2.1 Base de Dados

Todas as bases de dados utilizada nos experimentos possuem as variáveis descritas na Tabela 1.

Tabela 1: Variáveis consideradas e suas descrições.

Variável	Descrição
UF	Unidade Federativa onde o processo foi aberto (estado brasileiro).
Comarca	Comarca onde o processo foi aberto.
Juiz	Juiz responsável pelo caso.
Vara	Vara onde o processo foi aberto
Valor da Ação	Valor monetário pedido na ação.
Decisão Liminar	Decisão judicial sobre o pedido liminar (se houver).
Revelia	Houve ou não revelia (variável binária).
Tempo de Inatividade	Tempo desde a última atualização do processo.
Assunto	Assunto referente ao processo.
Classe do processo segundo CNJ	Classe do processo de acordo com classificação definida pelo CNJ

Como mencionado anteriormente, um total de 61 bases foram criadas com os mais de 600 mil processos. Estas bases não são disjuntas, tampouco cobrem todos os processos extraídos dos tribunais se somadas. Entre estas 61 bases, uma contém todos os processos extraídos, 21 possuem os processos separados por estado, em 16 delas foram separados os processos pelas comarcas das capitais de boa parte dos estados de todas as regiões do país, e em 23 delas os processos foram agrupados por vara, sendo utilizadas as maiores varas de 3 capitais brasileiras: São Paulo, Rio de Janeiro e Fortaleza. Todos os outros estados, comarcas e varas não foram contemplados neste trabalho por não haver volume acima de 1000 processos, ou por apresentar um desbalanceamento muito elevado na distribuição das sentenças.

Para estes experimentos, foram selecionados processos sentenciados como pedidos *procedentes*, e *improcedentes*. A rotulação para construção da base de dados foi realizada de maneira automática, através do uso de expressões regulares que são capazes de encontrar sentenças dentro dos andamentos de processos sentenciados. Finalmente, as sentenças foram agrupadas em dois grupos: *vitória* do réu (*i.e.* improcedente) e *derrota* do réu (*i.e.i* procedente e parcialmente procedente), o que caracteriza um problema de classificação binária.

As distribuições dos rótulos *vitória* e *derrota* em cada base e os percentuais de dados nulos por variável estão disponíveis em https://drive.google.com/file/d/1OEryYfM3g6WM5cW5s5ldIfeWUREqvYpz/view?usp=sharing. Devido ao tamanho da tabela com os dados mencionados, foi optado por manter tais informações em fonte externa.

Ao visitar o link mencionado, observa-se que existe um desbalanceamento em todas as bases entre os rótulos: os processos vitoriosos estão em menor número em todas as bases. Para solucionar o desbalanceamento nas bases selecionadas, foi utilizado *oversampling* aleatório, de maneira que a classe minoritária se iguala à classe majoritário em número de instâncias. Neste método, elementos da classe minoritária são aleatoriamente selecionados e replicados na base, até que os números de elementos em ambas as classes sejam igualados.

É importante lembrar que essas distribuições podem não corresponder ao quadro real, se forem considerados todos os processos existentes no país. Esta é composta apenas de alguns processos abertos contra seguradoras, apesar de base cobrir todas as regiões do Brasil.

2.2 Pré-Processamento e *Feature Engineering*

Para a construção do modelo, os dados passaram por processos de limpeza, criação e seleção de *features*. Durante a limpeza, algumas colunas categóricas passaram por um processo de padronização de valores, onde erros de grafia e variações semanticamente equivalentes foram eliminadas, além do agrupamento de valores pouco frequentes para uma categoria "outros". As definições dos limiares para estes agrupamentos dependeu da distribuição de cada *feature*.

Durante a criação de *features*, variáveis categóricas foram transformadas em números inteiros. Porém, devido ao fato de algoritmos baseados em florestas de árvores de decisão terem sido escolhidos para a construção dos modelos, o que será detalhado em breve, não foi utilizado o método *one-hot encoding* para codificação destas *features*. Além disso, pelo mesmo motivo, as variáveis

DOI: 10.25286/repa.v6i3.1725

numéricas não foram normalizadas, visto que, em algoritmos baseados em árvores de decisão, valores em diferentes escalas não interferem no processo de construção do modelo.

Duas variáveis categóricas apresentaram um grande volume de valores possíveis: o assunto do processo, que descreve o teor do mesmo, e a classe do processo segundo o Conselho Nacional de Justiça (CNJ), que também carrega um pouco da semântica do seu conteúdo. Para estes casos, visando facilitar o processo de aprendizagem do algoritmo, as variáveis foram consideradas do tipo 'texto' e foram codificadas utilizando o método *Latent Dirichlet Allocation* (LDA) **[18]**.

O método LDA é utilizado para modelagem de tópicos em bases de documentos. Neste método, uma matriz de frequência de termos por documento da base é decomposta em duas matrizes: uma que atribui um valor de representatividade dos termos por cada tópico e uma matriz que representa o quanto cada documento é representado por cada tópico. Em outras palavras, o método LDA pode ser considerado um método de agrupamento difuso, onde cada documento pertence a um dos 'n' tópicos em um certo grau. Portanto, cada documento pode ser codificado por um vetor, onde cada elemento do vetor representa o grau de pertencimento do documento em questão a um dos 'n' tópicos. Dessa forma, ao invés de representar cada documento por um vetor com 'N' elementos, onde 'N' é o número de termos pertencentes ao conjunto de documentos, como ocorreria com o método *Term Frequency-Inverse Document Frequency* (TF-IDF) **[19]**, por exemplo, cada documento pode ser codificado em um vetor de 'n' elementos, sendo n << N. Nestes experimentos, n = 20, resultando em 40 *features* para codificação do assunto e da classe do processo através do método LDA.

Para a seleção de *features*, as colunas com mais de 30% dos seus valores nulos em cada base foram excluídas. Os valores nulos das *features* restantes foram mantidos, devido à capacidade do algoritmo escolhido para construção de modelos lidar com tais valores, como será detalhado em seguida. Para os valores nulos das variáveis que determinam o assunto e a classe do projeto de acordo com o CNJ, as suas respectivas linhas foram removidas, pois tais variáveis, como mencionado anteriormente, devem gerar uma quantidade significativa de *features* através do método LDA, o que resultaria em uma proporção elevada de *features* em branco nestas linhas. Esta etapa gerou variações nos conjuntos de *features* selecionadas entre as diferentes bases utilizadas nestes experimentos.

2.3 Modelagem

Para a construção do modelo, foram utilizados algoritmos baseados em Florestas de Árvores de Decisão. Foram utilizadas as implementações disponíveis na biblioteca *lightgbm* para a linguagem de programação *Python* **[20]**. A escolha de Florestas de Árvores de Decisão se deu pelo enorme êxito obtido por essas técnicas em diversas competições na área onde as bases possuiam variáveis categóricas como parte considerável das suas colunas.

Árvores de Decisão são classificadores baseados em uma estrutura de árvore construída a partir das amostras de uma base de treinamento através de um algoritmo de aprendizagem de máquina **[21]**. *Ensembles* de classificadores são abordagens que utilizam um conjunto de classificadores para realizar, em conjunto, uma única tarefa de classificação. Florestas de Árvores de Decisão, como o nome já indica, são *ensembles* de Árvores de Decisão. Como é esperado de qualquer *ensemble* de classificadores, em comparação às Árvores de Decisão, as Florestas são modelos que possuem uma maior precisão, visto que diversos classificadores que, idealmente, possuem "visões" diferentes sobre um mesmo fenômeno representado por uma amostra de treinamento "opinam" sobre uma mesma instância a ser classificada, sendo a decisão final tomada em conjunto **[22]**.

É importante destacar que, nas implementações disponíveis na biblioteca *lightgbm*, cada nó possui um ramo padrão para os casos onde o valor do atributo correspondente é nulo, o que o torna capaz de lidar com *features* com valores nulos, como mencionado anteriormente. Além disso, a biblioteca disponibiliza múltiplas implementações de *Gradient Boosting Decision Trees* (GBDT): GBDT tradicional **[23]**, *Gradient-based One-Side Sampling* (GOSS) **[20]** e *Dropouts meet Multiple Additive Regression Trees* (DART) **[24]**.

GBDTs são *ensembles* de Árvores de Decisão onde diversos classificadores fracos são construídos em sequência, onde cada classificador é formado a partir de uma base ponderada onde os elementos para os quais o classificador anterior apresentou maior erro possuem maior peso. Estes algoritmos constroem um

conjunto de classificadores fracos, os quais possuem uma "visão simplista" do espaço de decisão. Porém, se usados em conjunto podem superar classificadores únicos e mais complexos **[22].**

Para a seleção do algoritmo e de seus hiperparâmetros, utilizamos o método *Population Based Training* (PBT) **[25]**, o qual herda conceitos de *Random Search* e Computação Evolucionária. Nesta abordagem uma população de *workers* iniciam os treinamentos de diferentes modelos com seus respectivos parâmetros e hiperparâmetros. Após um determinado tempo, avalia-se se houve melhora significativa no modelo durante as últimas iterações. Caso negativo, os parâmetros e os hiperparâmetros do modelo em questão são substituídos pelos valores do melhor modelo na população. Após a cópia, com uma determinada probabilidade, os hiperparâmetros copiados são "perturbados" aleatoriamente, gerando um novo conjunto de valores, o qual será usado para evoluir os parâmetros recém copiados. Este método apresenta as seguintes vantagens: cada *worker* pode ser executado de maneira independente e assíncrona dos outros, o que permite grandes *speedups* se o processo for paralelizado; as regiões do espaço de hiperparâmetros mais promissoras recebem um maior número de recursos para serem exploradas, acelerando o processo de convergência para boas regiões; os hiperparâmetros são ajustados dinamicamente, o que permite a busca por bons procedimentos de *annealing*, ou seja, boas políticas de controle de hiperparâmetros.

A Tabela 2 descreve cada uma das configurações geradas para serem usadas pelo algoritmo PBT na seleção de modelos e hiperparâmetros. Nesta tabela, cada coluna representa uma configuração inicial, que é executada por um *worker*. Para cada U(a,b), um valor inteiro entre 'a' e 'b' é gerado aleatoriamente a partir de uma distribuição probabilística uniforme, enquanto LU(a,b) gera um valor entre 'a' e 'b' através de uma distribuição log-uniforme. É importante observar que três métodos são testados 32 vezes cada: GBDT, DART e GOSS. Com as funções U e LU, cada um dos *workers* é iniciado com um valor aleatório para cada um dos hiperparâmetros listados, ou seja, cada um dos três métodos é inicializado com hiperparâmetros com valores diferentes. As perturbações ocorrem através das mesmas funções definidas na tabela. A execução do método termina quando nenhum dos *workers* apresentam mais melhorias significativas.

Para avaliar o método descrito, cada base foi dividida em três partes: base de treinamento, base de validação e base de teste. A base de treinamento foi utilizada para a construção do modelo em si, enquanto a base de validação foi utilizada para avaliar a qualidade de um determinado modelo pelo método PBT de seleção de hiperparâmetros e modelos. Por fim, após tal seleção, a base de teste foi utilizada para obter a qualidade do modelo em uma base não vista pelos algoritmos de treinamento e seleção de hiperparâmetros e modelos, eliminando as chances de *information leakage* entre os processos de treinamento e teste. No particionamento, cada base foi ordenada crescentemente de acordo com a data de distribuição (*i.e.* abertura) dos processos, sendo os 60% mais antigos processos separados para a base de treinamento, os 20% seguintes para a base de validação e, por fim, os 20% mais recentes para a base de teste. Desta forma, podemos avaliar a capacidade dos modelos realizarem previsões acerca de processos futuros a partir de processos passados.

Tabela 2: Configurações dos algoritmos de treinamento para serem usadas pelo método PBT para treinamento e controle de hiperparâmetros.

Hiperparâmetro	**1**	**2**	**3**
Boosting	GBDT	DART	GOSS
# Leaves	U(32, 4098)	U(32, 4098)	U(32, 4098)
Learning Rate	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)
Max bin	U(32, 1024)	U(32, 1024)	U(32, 1024)
Max depth	U(5, 12)	U(5, 12)	U(5, 12)
Min data in leaf	U(5, 100)	U(5, 100)	U(5, 100)
L1 Coefficient	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)
L2 Coefficient	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)
Hiperparâmetro	**4**	**5**	**6**
Boosting	GBDT	DART	GOSS
# Leaves	U(32, 4098)	U(32, 4098)	U(32, 4098)
Learning Rate	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)
Max bin	U(32, 1024)	U(32, 1024)	U(32, 1024)
Max depth	U(5, 12)	U(5, 12)	U(5, 12)
Min data in leaf	U(5, 100)	U(5, 100)	U(5, 100)
L1 Coefficient	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)
L2 Coefficient	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)
…	…	…	…

DOI: 10.25286/repa.v6i3.1725

Hiperparâmetro	94	95	96
Boosting	GBDT	DART	GOSS
# Leaves	U(32, 4098)	U(32, 4098)	U(32, 4098)
Learning Rate	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)
Max bin	U(32, 1024)	U(32, 1024)	U(32, 1024)
Max depth	U(5, 12)	U(5, 12)	U(5, 12)
Min data in leaf	U(5, 100)	U(5, 100)	U(5, 100)
L1 Coefficient	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)
L2 Coefficient	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)	LU(1e-8, 1e-1)

Para cada experimento, obteve-se a F1 *score* ponderada (*i.e.* média da F1 *score* ponderada pela proporção das decisões - VITÓRIA ou DERROTA - de cada base de teste) e a acurácia de cada modelo nas bases de teste.

3 Resultados e Discussões

Na base completa, o melhor modelo encontrado obteve 0,66 de acurácia e F1 *score* ponderada. Com as bases de processos agrupados por estado, as previsões para os estados do Acre e do Mato Grosso do Sul obtiveram Acurácia e F1 ponderada abaixo de 0,5, o que pode ser considerado um resultado insatisfatório. Para o estados de Goiás, Minas Gerais, Paraíba, Pernambuco, Santa Catarina e Sergipe, os melhores modelos não ultrapassaram 0,6 de acurácia ou F1 ponderada. Apenas os modelos gerados a partir das bases do estados de Alagoas, Ceará, Espírito Santo e Maranhão obtiveram resultados acima de 0,7. Os 9 estados restantes obtiveram obtiveram de 0,6 a 0,7 de acurácia e F1 ponderada. Estes resultados são detalhados na Tabela 3.

Tabela 3: Resultados dos experimentos com as bases separadas por estado.

Base	Acurácia	F1 ponderada
Estado do Acre	0,36	0,35
Estado de Alagoas	0,7	0,7
Estado da Bahia	0,62	0,61
Estado do Ceará	0,75	0,75
Distrito Federal	0,62	0,63
Estado do Espírito Santo	0,71	0,72
Estado de Goiás	0,59	0,6
Estado do Maranhão	0,74	0,75
Estado de Minas Gerais	0,55	0,55
Estado do Mato Grosso	0,63	0,63
Estado do Mato Grosso do Sul	0,45	0,46
Estado da Paraíba	0,59	0,59
Estado de Pernambuco	0,57	0,58
Estado do Paraná	0,61	0,61
Estado do Rio de Janeiro	0,66	0,59
Estado do Rio Grande do Norte	0,67	0,65
Estado de Rondônia	0,62	0,62
Estado do Rio Grande do Sul	0,65	0,65
Estado de Santa Catarina	0,59	0,58
Estado de Sergipe	0,55	0,55
Estado de São Paulo	0,68	0,65

Na Tabela 4, detalham-se os resultados obtidos com as bases com processos agrupados pela comarca. Neste caso, apenas o modelo gerado com a base de Florianópolis obteve acurácia e F1 ponderada abaixo de 0,5. Belo Horizonte, Maceió, Curitiba e Rio Branco apresentaram para ambas as métricas valores entre 0,5 e 0,6, enquanto Cuibá, Fortaleza, Porto Alegre e São Luis ultrapassaram a marca de 0,7. Em todo o restante, i.e. em 7 bases das 16, nossos modelos permaneceram entre 0,6 e 0,7 em ambas as métricas.

Tabela 4: Resultados dos experimentos com as bases separadas por comarca.

Base	Acurácia	F1 ponderada
Comarca de Belo Horizonte	0,52	0,52
Comarca de Cuiabá	0,73	0,74
Comarca de Curitiba	0,55	0,53
Comarca de Florianópolis	0,49	0,5
Comarca de Fortaleza	0,77	0,75
Comarca de Goiânia	0,65	0,61
Comarca de Maceió	0,55	0,58
Comarca de Natal	0,6	0,6
Comarca de Porto Alegre	0,72	0,66
Comarca de Recife	0,68	0,64
Comarca de Rio Branco	0,5	0,51
Comarca do Rio de Janeiro	0,67	0,58
Comarca de Salvador	0,68	0,7
Comarca de São Luis	0,73	0,76
Comarca de São Paulo	0,69	0,68
Comarca de Vitória	0,61	0,62

Com o objetivo de comparar os modelos gerados a partir de bases de processos agrupados por estado com as bases agrupadas por comarca, sendo que 5 dos 21 estados não tiveram suas capitais contempladas nestas bases, aplicamos o teste de hipótese de Wilcoxon com 5% de significância nas distribuições dos valores de acurácia e F1 ponderada

de ambos os grupos de base. A hipótese inicial era de que teríamos maior consistência nas decisões dos juízes de um conjunto menor de tribunais, ou seja, conseguiríamos modelos melhores por termos bases mais fáceis se a construção dos modelos fossem concentrados em bases mais restritas. Porém, apesar da necessidade de ampliarmos as bases separadas por comarca com mais cidades de cada estado, não obtivemos nenhuma evidência de que exista qualquer ganho significativo ao aumentar a granularidade dos agrupamentos de estados para cidades ao montar as bases de treinamento, visto que os p-valores para as distribuições da acurácia e da F1 ponderada foram de 0,6 e 0,57, respectivamente.

Na Tabela 5, encontram-se detalhados os resultados dos experimentos para as bases com processos agrupados pela vara. Como concentramos as varas em apenas 3 comarcas, como mencionado anteriormente, não podemos realizar nenhuma comparação entre estes resultados e os obtidos com as bases separadas por comarca e por estado. Porém, observa-se resultados no mesmo patamar de todas as outras bases. No entanto, podemos destacar os resultados alcançados nas 12ª, 24ª e 30ª Varas Cíveis de Fortaleza, nas quais os nossos modelos atingiram valores próximos a 0,8 na acurácia e F1 ponderada.

Tabela 5: Resultados dos experimentos com as bases separadas por vara.

Base	Acurácia	F1 ponderada
1º Juizado Especial Cível de São Paulo	0,52	0,53
2º Juizado Especial Cível de São Paulo	0,53	0,53
1ª Vara Cível de São Paulo	0,76	0,73
2ª Vara Cível de São Paulo	0,72	0,72
3ª Vara Cível de São Paulo	0,67	0,64
4ª Vara Cível de São Paulo	0,66	0,66
5ª Vara Cível de São Paulo	0,7	0,7
6ª Vara Cível de São Paulo	0,65	0,64
7ª Vara Cível de São Paulo	0,6	0,62
8ª Vara Cível de São Paulo	0,57	0,5
9ª Vara Cível de São Paulo	0,68	0,66
10ª Vara Cível de São Paulo	0,65	0,67
11ª Vara Cível de São Paulo	0,65	0,56
12ª Vara Cível de Fortaleza	0,77	0,78
14ª Vara Cível de Fortaleza	0,73	0,73
24ª Vara Cível de Fortaleza	0,79	0,77
30ª Vara Cível de Fortaleza	0,77	0,76
1ª Vara Cível do Rio de Janeiro	0,69	0,64
2ª Vara Cível do Rio de Janeiro	0,57	0,57
3ª Vara Cível do Rio de Janeiro	0,55	0,56
4ª Vara Cível do Rio de Janeiro	0,55	0,55
5ª Vara Cível do Rio de Janeiro	0,64	0,63
6ª Vara Cível do Rio de Janeiro	0,7	0,68

4 Conclusões

Este artigo apresenta modelos de previsão de sentenças jurídicas de tribunais brasileiros de primeira instância construídos a partir de 61 bases de dados provenientes de todas as regiões do Brasil, totalizando mais de 600 mil processos. Para a construção destas bases, foram utilizados apenas dados disponíveis publicamente nos sites dos tribunais. Para a construção destes modelos, foram utilizados *ensembles* de Árvores de Decisão, cujos hiperparâmetros foram ajustados pelo método *Population Based Training*.

Após os experimentos realizados chegou-se às seguintes conclusões:

- Como pôde ser visto nos experimentos, mesmo usando métodos do estado-da-arte na área, as acurácias variaram de 36% a 79%, o que demonstra uma grande variância de complexidade entre as bases disponíveis publicamente nos tribunais do Brasil, no que se refere à previsibilidade da sentença de primeira instância.
- Não encontramos evidências de que podemos obter modelos mais precisos se aumentarmos a granularidade do agrupamento das bases, construindo modelos mais focados em recortes mais especializados em locais específicos.
- Os tribunais brasileiros de primeira instância apresentam uma deficiência de informação útil disponível publicamente que possa ser utilizada para prever decisões desta esfera. Observa-se principalmente falta de informações detalhadas acerca do teor do processo, pois o assunto e a classe do processo segundo o CNJ apresentam tais informações de forma altamente resumida, sem os detalhes que normalmente embasam as decisões dos juízes. É importante destacar que tais informações podem ser encontradas em petições iniciais. Porém, tais documentos não se encontram disponíveis ao público. Portanto, para permitir uma maior previsibilidade das

DOI: 10.25286/repa.v6i3.1725

sentenças de primeiro grau a partir dos dados públicos, os tribunais devem passar a fornecer dados de melhor qualidade, *i.e.* mais corretos e mais completos.

5 Agradecimentos

Os autores deste estudo agradecem à Fundação de Amparo a Ciência e Tecnologia de PE (FACEPE) e à empresa Intelivix LTDA por terem financiado este projeto.

Referências

[1] ALETRAS, N. et al. Predicting judicial decisions of the european court of human rights: a natural language processing perspective. PeerJ Computer Science, v. 2, p. e93,2016.

[2] LIU, Z.; CHEN, H. A predictive performance comparison of machine learning models forjudicial cases. In:2017 IEEE Symposium Series on Computational Intelligence (SSCI). 2017. p. 1–6.

[3] MEVDEVA, M. V. M.; WIELING, M. Judicial decisions of the european court of human rights: looking into the crystal ball. In: Proceedings of the Conference on Empirical Legal Studies in Europe, 2018.

[4] CHALKIDIS, I.; ANDROUTSOPOULOS, I.; ALETRAS, N. Neural legal judgment prediction in English. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics,2019. p. 4317–4323.

[5] KAUR, A. Convolutional Neural Network-based Automatic Prediction if Judgments if The European Court of Human Rights. Dissertação (Mestrado) — Technological University Dublin, 2019.

[6] QUEMY, A. European court of human right open data project. ArXiv,abs/1810.03115, 2018.

[7] O'SULLIVAN, C.; BEEL, J. Predicting the outcome of judicial decisions made by the european court of human rights. ArXiv, abs/1912.10819, 2019.

[8] XIAO, C. et al. Cail 2018: A large-scale legal dataset for judgment prediction. ArXiv,abs/1807.02478, 2018.

[9] YANG, W. et al. Legal judgment prediction via multi-perspective bi-feedback network. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Aug 2019.

[10] LI, S. et al. Mann: A multichannel attentive neural network for legal judgment prediction. IEEE Access, v. 7, p. 151144–151155, 2019.

[11] CHAO, W.-H. et al. Interpretable charge prediction for criminal cases with dynamicrationale attention. J. Artif. Intell. Res., v. 66, p. 743–764, 2019.

[12] KOWSRIHAWAT, K.; VATEEKUL, P.; BOONKWAN, P. Predicting judicial decisions of criminal cases from thai supreme court using bi-directional GRU with attention mechanism. In: 2018 5th Asian Conference on Defense Technology (ACDT). 2018. p. 50–55.ISSN.

[13] VIRTUCIO, M. B. L. et al. Predicting decisions of the philippine supreme court usingnatural language processing and machine learning. In:2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). 2018. v. 02, p.130–135. ISSN 0730-3157.

[14] WALTL, B. et al. Predicting the outcome of appeal decisions in germany's tax law. In: International Conference on Electronic Participation. 2017.

[15] KAUFMAN, A. R.; KRAFT, P.; SEN, M. G. Improving supreme court forecasting using boosted decision trees. In: 9th International Conference on Electronic Participation. 2019.

[16] QUEUDOT, M.; MEURS, M.-J. Artificial intelligence and predictive justice: Limitations and perspectives. In: IEA/AIE, 2018.

[17] LAGE-FREITAS, A. et al. Predicting brazilian court decisions. ArXiv,abs/1905.10348, 2019.

[18] BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. J. Mach. Learn.Res., JMLR.org, v. 3, n. , p. 993–1022, mar. 2003. ISSN 1532-4435.

[19] ROBERTSON, S. E. Understanding inverse document frequency: on theoretical arguments for idf. Journal of Documentation, v. 60, p. 503–520, 2004.

[20] KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. In: NIPS.b 2017.

[21] MURTHY, S. K. Automatic construction of decision trees from data:Amulti-disciplinary survey.Data Min. Knowl. Discov., Kluwer Academic Pu-blishers, USA, v. 2, n. 4, p. 345–389, dez. 1998. ISSN 1384-5810.

[22] DIETTERICH, T. G. Ensemble methods in machine learning. In: Proceedings of the First International Workshop on Multiple Classifier Systems. Berlin, Heidelberg:Springer-Verlag, 2000. (MCS '00), p. 1–15. ISBN 3540677046.

[23] BREIMAN, L. Arcing the edge. 1997.

[24] RASHMI, K. V.; GILAD-BACHRACH, R.DART: Dropouts meet Multiple Additive Regression Trees. ArXiv,abs/1505.01866. 2015.

[25] JADERBERG, M. et al. Population Based Training of Neural Networks. ArXiv,abs/1711.09846. 2017.

DOI: 10.25286/repa.v6i3.1725