

# Projeto Final

## Ciência de Dados

Membros: Ricardo Cavalcante - 377744  
Tales Araujo - 374953  
Lucas S. Fernandes - 486265  
Rômulo Férrer Filho - 385218  
João Pedro B. Andrade - 486454





## Motivações - 0 problema

- Classificar documentos de legislação da SEFAZ, disponíveis no portal Sefaz Legis, de acordo como seu tipo: Lei, Decreto, Instrução Normativa, etc.
  - o título do documento contém o tipo documental e deverá ser usado como label;
  - os trabalhos serão avaliados pela acurácia do modelo e pelo tratamento dado aos dados, na solução.

# Aquisição dos dados

- A fonte dos dados é feita com uma ferramenta chamada alfresco, que dificultou a navegação automatizada.
- 160 pastas 14 externas e 146 internas
- Script para recuperar itens das pastas:

```
let links = document.getElementsByTagName('a');  
  
for(let i=0; i < links.length; i++) {  
  links[i].dispatchEvent(new MouseEvent("click"))  
}
```

- 3612 arquivos



# Pré-processamento

- Extração do conteúdo dos PDFs
  - Py2PDF
  - Slate
  - PdfPlumber
- Erros de Encoding
- Filtros de classe:
  - Ato Declaratório
  - Decreto
  - Instrução Normativa
  - Lei
  - Norma de Execução
  - Nota Explicativa

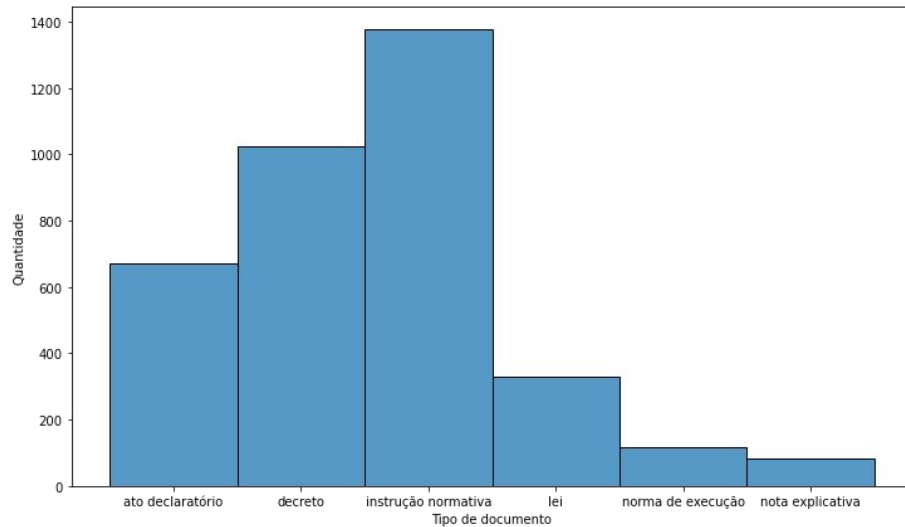




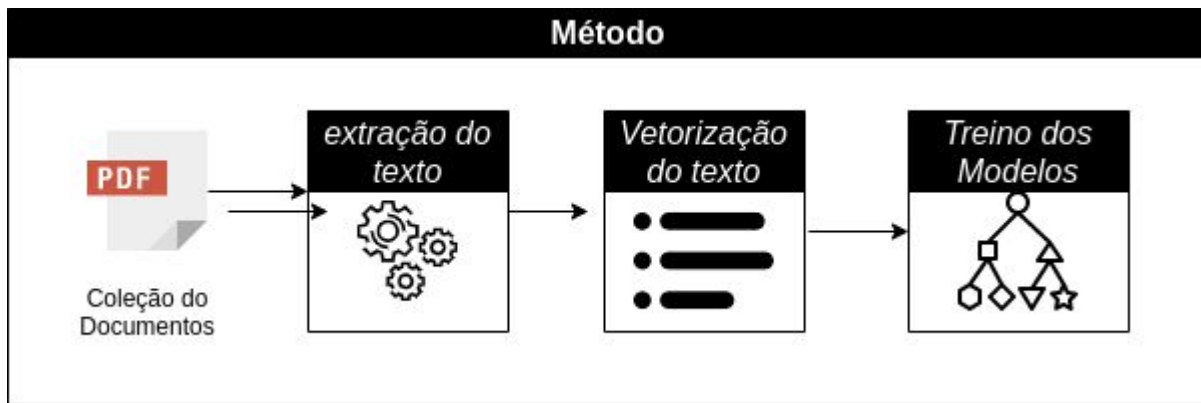
# Análise dos dados

Questões encontradas:

- Dados faltantes
- Ambiguidades nas *labels*
- Conjuntos não-disjuntos de classes
- Seleção de classes para construção do modelo



# Solução



O conjunto de métodos abaixo foi aplicado para o dataset e os em destaque obtiveram melhor resultado, portanto foram escolhidos como solução principal:

- **Count Vectorizer**
- **Decision Tree**
- TF-IDF Vectorizer
- Naive Bayes
- Random Forest

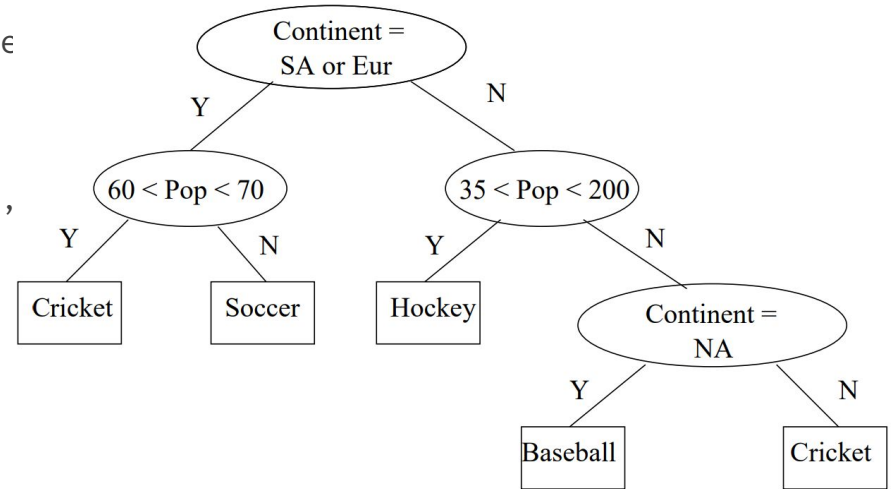


# Count-Vectorizer

- Tokenização de palavras via *Bag-of-Words* em coleção de documentos
- Simplificação de sentenças por remoção de termos prolixos
- Retorna vetor codificado contendo:
  - Comprimento do Vocabulário
  - Contagem de cada palavra

# Decision Tree

- Utiliza propriedades do vetor de atributos para decidir a classe da entrada.
- Representada em forma de árvore, onde cada nó é um teste e as folhas são as decisões.



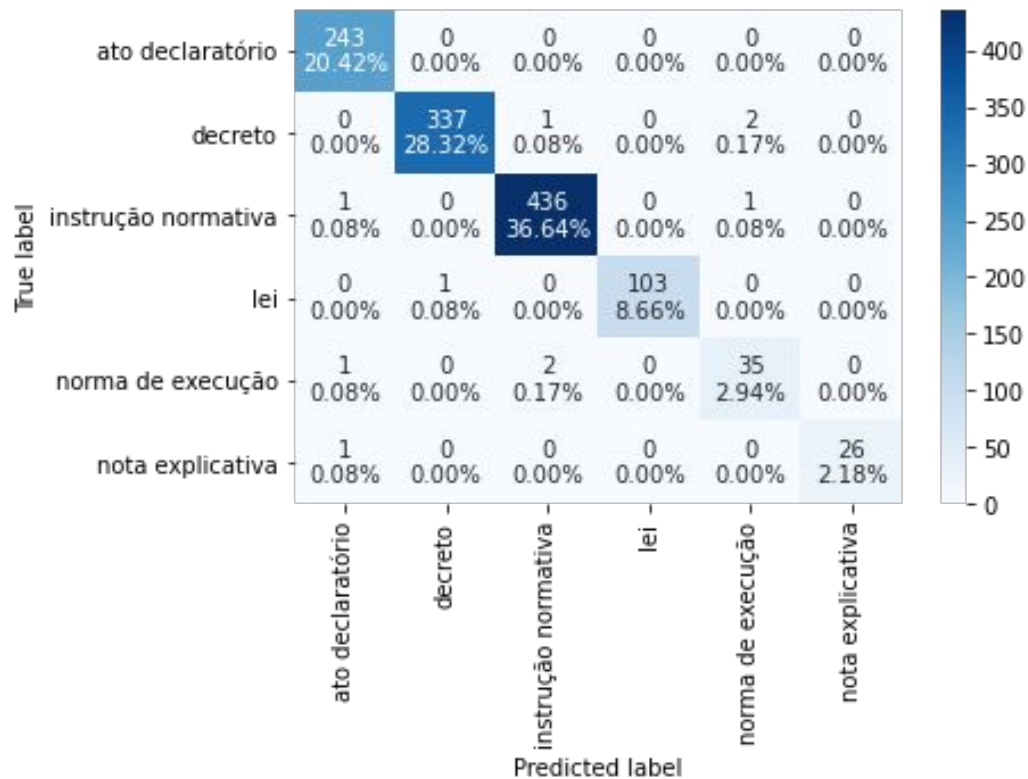
Exemplo de DT para esporte favorito de um país. Fonte: MMDS (Leskovec *et. al.*)



# Resultados

	precision	recall	f1-score	support
ato declaratório	0.99	1.00	0.99	243
decreto	1.00	0.99	0.99	340
instrução normativa	0.99	1.00	0.99	438
lei	1.00	0.99	1.00	104
norma de execução	0.92	0.92	0.92	38
nota explicativa	1.00	0.96	0.98	27
accuracy			0.99	1190
macro avg	0.98	0.98	0.98	1190
weighted avg	0.99	0.99	0.99	1190

# Resultados



Accuracy=0.992



# Conclusão

- A combinação *CountVectorizer* e *DecisionTree* demonstrou excelentes resultados
- Possíveis próximos passos:
  - Validação cruzada
  - Tratar o desbalanceamento dos dados (resampling?)
  - Testar outras técnicas de vetorização de texto e classificadores



## Dúvidas?

Obrigado pela atenção!

Repositório do Trabalho:

[Github - Trabalho Sefaz](#)



Observação: Trabalho foi atualizado (notebook link no github) com Cross-Validation para ENTREGA FINAL como sugerido no checkpoint 3