

# **Author Name Disambiguation (AND) usando TF-IDF e RandomForestClassifier**

Apresentação sobre o processo de desambiguação de nomes de autores, uso de vetorização de texto (TF-IDF), RandomForestClassifier e a avaliação de desempenho do modelo.

## **Alunos:**

- **Natan Rodrigues**
- **Leonardo Seger**

# O que é AND (Author Name Disambiguation)?

- **Definição:** A desambiguação de nomes de autores – Author Name Disambiguation (AND) visa resolver problemas de ambiguidade, onde diferentes autores podem ter o mesmo nome ou nomes semelhantes em bases de dados bibliográficas.
- **Objetivo:** Diferenciar corretamente autores distintos com base em suas publicações, coautores e metadados, associando o nome correto ao autor em questão.
- **Desafio:** Autores com nomes idênticos ou similares criam incerteza sobre a verdadeira identidade nas bases de dados bibliográficas. A combinação de técnicas de processamento de texto e machine learning ajuda a resolver esse desafio.



# Dataset utilizado

- **Fonte:** - O dataset utilizado contém publicações científicas do AMiner, incluindo metadados sobre cada publicação e seus autores. <https://www.aminer.cn/disambiguation>
- **Estrutura do dataset:**
  - `author\_id`: Identificação única do autor.
  - `author\_name`: Nome do autor.
  - `coauthors`: Lista de coautores em cada publicação.
  - `publication\_title`: Título da publicação.
  - `abstract`: Resumo da publicação.
  - `venue`: Conferência ou local onde a publicação foi apresentada.
- 109 Referências ambíguas de nomes
- 7447 artigos de 1546 autores únicos
- **Limpeza de dados:** - Tratamos valores ausentes, substituindo por `"No Data"` para garantir que o modelo não receba valores nulos que poderiam prejudicar o treinamento.

author_id	author_name	coauthors	publication_title	abstract	venue
5a33917406df07e704c7afd5	Barry Wilkinson	Mark A. Holliday, Barry Wilkinson, James Ruff	Using an end		
5a33917406df07e704c7afd5	Barry Wilkinson	Rahman Tashakkori, Barry L. Kurtz, Barry Wilkinson, Mark A.			
5a33917406df07e704c7afd5	Barry Wilkinson	Oscar Ardaiz-Villanueva, Miguel L. Bote-Lorenzo, Amy W. Ap			
5a33917406df07e704c7afd5	Barry Wilkinson	Barry Wilkinson, Mark A. Holliday, Clayton Ferner	Experien		
5a33917406df07e704c7afd5	Barry Wilkinson	Barry Wilkinson, Tanusree Pai, Meghana Miraj	A Distributed		
5a33917406df07e704c7afd5	Barry Wilkinson	Kenneth E. Hoganson, Barry Wilkinson, W. Homer Carlisle	Ap		
5a33917406df07e704c7afd5	Barry Wilkinson	Mark A. Holliday, Barry Wilkinson, Jeffrey House, Samir Dao			
5a33917406df07e704c7afd5	Barry Wilkinson	Jens Mache, Amy W. Apon, Thomas Feilhauer, Barry Wilkinson			
5a33917406df07e704c7afd5	Barry Wilkinson	Barry Wilkinson, Clayton Ferner	Towards a top-down approa		
5a33917406df07e704c7afd5	Barry Wilkinson	Jeremy F. Villalobos, Barry Wilkinson	Latency hiding by r		



# Codificação dos autores e separação dos dados

- Codificação do `author\_id`:
- O `LabelEncoder` do scikit-learn foi utilizado para transformar o `author\_id` em classes numéricas, que são necessárias para que o modelo possa realizar a previsão.
- Separação dos dados: - Dividimos o dataset em duas partes:
  - Treino (70%) : Para que o modelo aprenda os padrões.
  - Teste (30%) : Para avaliar o desempenho do modelo em dados não vistos.

# O que é TF-IDF?

- **TF-IDF** significa "Term Frequency-Inverse Document Frequency". Transforma texto em representações numéricas, calculando o peso de cada termo com base na sua frequência no documento (TF) e na raridade no conjunto de documentos (IDF).
- **Por que foi utilizado TF-IDF?:** - O TF-IDF foi utilizado para capturar a relevância dos termos em relação ao conjunto de documentos.
  - } Em problemas como o AND, é importante identificar quais palavras e nomes de coautores são mais informativos para distinguir entre diferentes autores. - O uso de TF-IDF permite transformar texto em uma forma que o modelo de machine learning pode processar, eliminando a necessidade de manipular diretamente strings. Ele também ajuda a filtrar termos menos relevantes, como palavras comuns, que podem ser menos úteis para a tarefa de classificação.
- **Aplicação no AND:** - Foi aplicado às colunas `publication\_title`, `coauthors`, `venue` e `abstract`.
- **Por que é útil?:** - A técnica de TF-IDF reduz o impacto de termos muito comuns, como palavras genéricas, ao dar mais peso a termos específicos e raros, que são mais úteis na diferenciação entre autores.



# O que é o Random Forest?

- O **RandomForestClassifier** é um algoritmo de aprendizado supervisionado baseado em múltiplas árvores de decisão. Ele cria várias árvores de decisão e combina suas previsões para produzir uma classificação mais robusta.
- **Por que foi utilizado o Random Forest?:** O Random Forest foi escolhido porque é um modelo capaz de lidar com problemas de classificação multiclasse, como o AND. O modelo também pareceu lidar bem com dados de alta dimensionalidade, como os vetores TF-IDF, onde há muitas features derivadas de texto.
- **No contexto do AND:**
  - O RandomForest foi utilizado para prever o `author\_id` (classe do autor) com base nas features TF-IDF extraídas de metadados textuais como coautores, títulos de publicações, venue e abstracts.
  - Como o problema de AND envolve múltiplas classes, o RandomForest é adequado por ser capaz de lidar com essas classificações complexas.

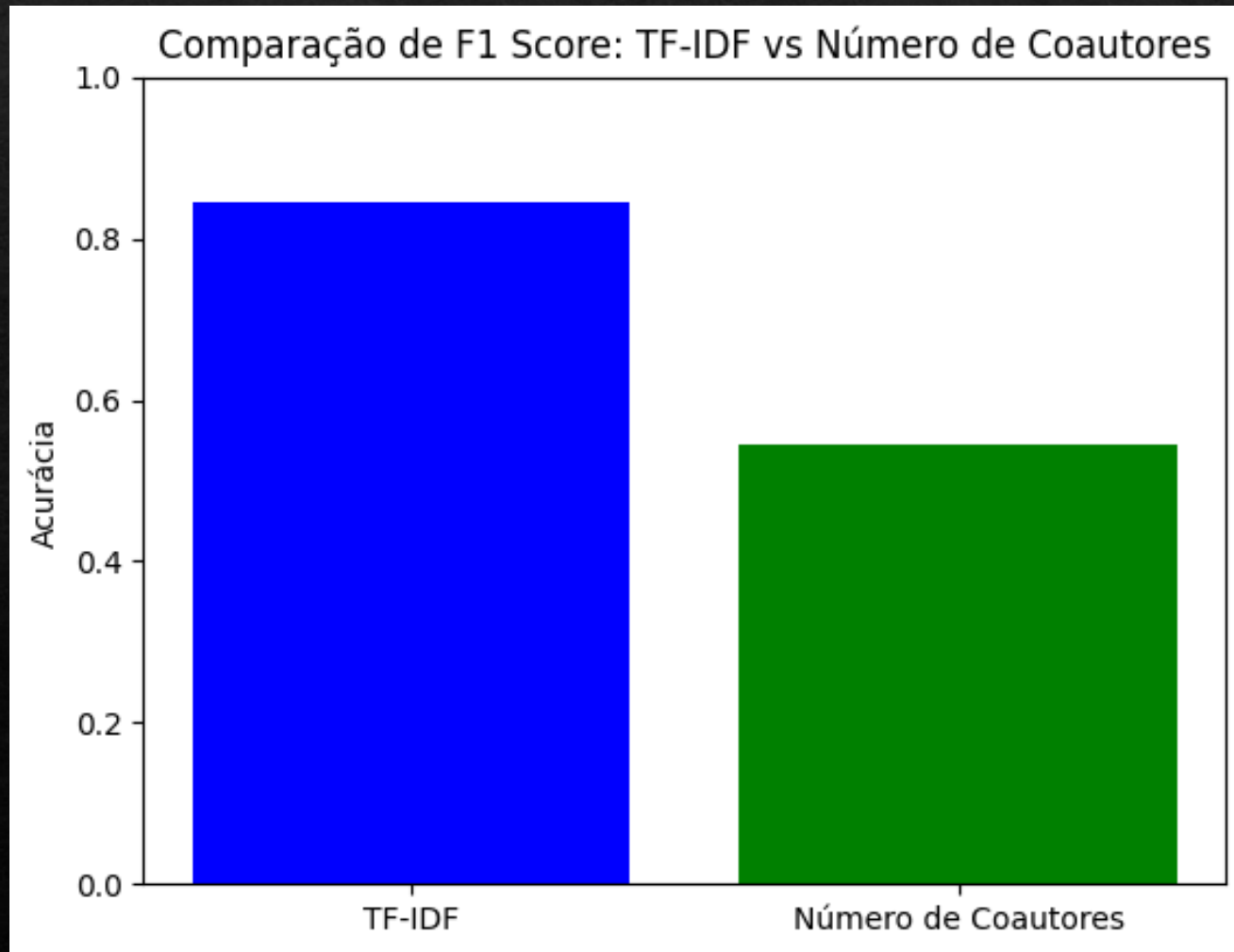
# Avaliação do modelo

- **Métricas de avaliação:**
- Precisão: Mede quantas das previsões positivas feitas pelo modelo estavam corretas.
- Recall: Mede quantas instâncias positivas reais foram corretamente identificadas pelo modelo.
- F1-Score: É uma combinação de precisão e recall, balanceando o impacto de falsos positivos e falsos negativos.

•

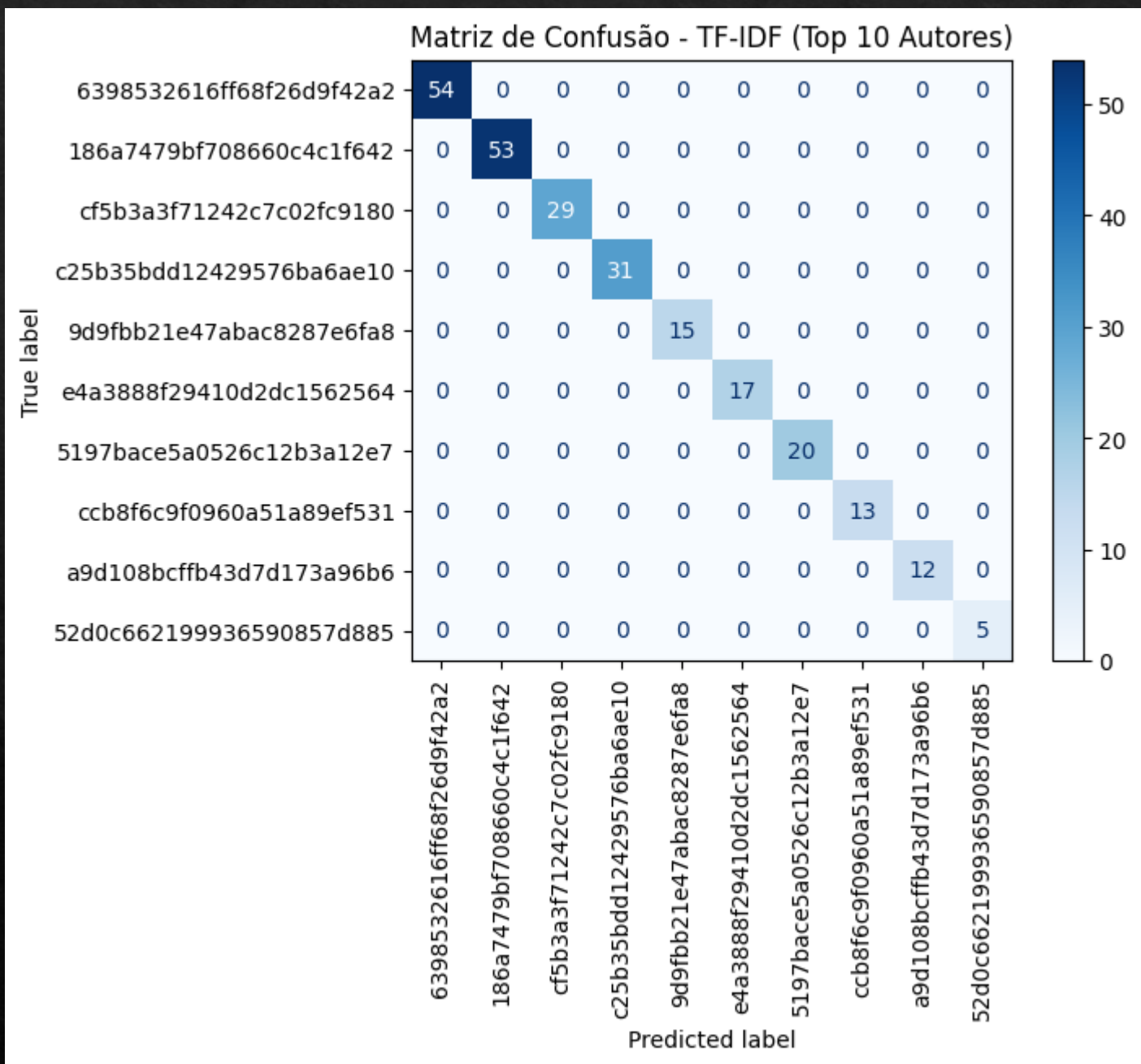


# Resultados



TF-IDF - F1-Score: 0.8439201451905626

Número de Coautores - F1-Score: 0.5444646098003629





# Conclusão

- Implementação:
  - Implementamos uma solução para o problema de desambiguação de nomes de autores (AND) usando vetorização TF-IDF e *RandomForestClassifier*.
- Desempenho:
  - O modelo apresentou bons resultados, demonstrando uma bom valor de F1 na tarefa de AND.
- Próximos passos:
  - Explorar outros modelos de machine learning, como redes neurais, e técnicas avançadas de pré-processamento de texto, como *embeddings* de palavras (Word2Vec ou BERT), para potencialmente melhorar a performance.
- Código disponível em: [https://github.com/natansr/and\\_tfidf\\_random\\_forest.git](https://github.com/natansr/and_tfidf_random_forest.git)