# Proposta de Projeto Final - Aprendizagem de Máquina (2025.2)

**Instituição:** Universidade Federal do Ceará (UFC)

**Professor:** César Lincoln Cavalcante Mattos

**Equipe:** Artur Garcia, Artur Saraiva

## 1. Título do Projeto

**Predição de Sucesso Eleitoral para Deputado Federal: Uma Abordagem Comparativa de Classificação Supervisionada em Dados Desbalanceados**

## 2. Definição do Problema

O sistema eleitoral brasileiro gera uma grande quantidade de dados públicos, permitindo a análise de quais fatores influenciam o sucesso de uma candidatura. O problema abordado neste projeto é de **Classificação Binária Supervisionada**.

O objetivo é treinar modelos de Aprendizagem de Máquina capazes de prever se um candidato ao cargo de **Deputado Federal** será **"Eleito" (1)** ou **"Não Eleito" (0)**, com base exclusivamente em atributos declarados ao Tribunal Superior Eleitoral (TSE) antes do pleito.

Este problema apresenta um desafio clássico de ML: o **desbalanceamento de classes**, visto que a proporção de candidatos eleitos é significativamente menor do que a de não eleitos (geralmente menos de 10% de sucesso).

## 3. Fonte de Dados (Dados Reais)

Utilizaremos os dados abertos disponibilizados pelo **Tribunal Superior Eleitoral (TSE)** através do Portal de Dados Abertos.

- **Fonte:** [Repositório de Dados Eleitorais do TSE](https://dados.gov.br/ ou https://www.google.com/search?q=https://sig.tse.jus.br/)
- **Recorte Temporal:** Eleições Gerais de 2022 (e possivelmente 2018 para aumento do dataset, se necessário).
- **Volume Estimado:** Aproximadamente 10.000 a 15.000 instâncias (candidatos) por eleição.

### 3.1. Atributos (Features) Previstos

As variáveis independentes (X) incluirão dados socioeconômicos e políticos:

1. **Dados Pessoais:** Idade, Gênero, Estado Civil, Grau de Instrução, Ocupação.
2. **Dados Financeiros:** Total de Bens Declarados (numérico), Teto de Gastos de Campanha.
3. **Dados Políticos:** Partido, Coligação (se houver), Situação de Reeleição (bool), Estado (UF).
4. **Target (Y):** Situação final (Eleito/Eleito por média/Eleito por QP = 1; Suplente/Não Eleito = 0).

## 4. Metodologia Proposta

#### 4.1. Pré-processamento

- **Limpeza:** Tratamento de valores nulos (ex: candidatos sem bens declarados).
- **Codificação:** Utilização de *One-Hot Encoding* para variáveis categóricas (Estado, Gênero) e *Target Encoding* para variáveis de alta cardinalidade (Partido, Ocupação).
- **Tratamento de Desbalanceamento:** Como a classe positiva (Eleito) é minoritária, aplicaremos técnicas como:
  - *Undersampling* da classe majoritária.
  - *SMOTE (Synthetic Minority Over-sampling Technique)*.
  - Ajuste de *Class Weights* nos algoritmos.

#### 4.2. Modelos Selecionados

Para atender ao requisito de comparação de desempenho, utilizaremos três abordagens distintas:

1. **Regressão Logística:** Servirá como *baseline* (modelo base) e permitirá alta interpretabilidade (analisar quais coeficientes/atributos pesam mais na vitória).
2. **Random Forest:** Modelo de *ensemble* robusto, capaz de capturar relações não-lineares e menos sensível a outliers.
3. **Gradient Boosting (XGBoost ou LightGBM):** Estado da arte para dados tabulares, focado em maximizar a performance preditiva.

#### 4.3. Estratégia de Avaliação

Dado o desbalanceamento, a Acurácia não será uma métrica confiável. A avaliação será feita utilizando **Validação Cruzada (k-fold cross-validation)** com $k = 5$ ou $k = 10$, observando as seguintes métricas:

- **F1-Score:** Média harmônica entre precisão e recall (crucial para classes desbalanceadas).
- **AUC-ROC:** Para avaliar a capacidade do modelo de distinguir entre as classes.
- **Matriz de Confusão:** Para visualizar falsos positivos e falsos negativos.

## 5. Cronograma Macro

- **09/12 a 15/12:** Coleta e Limpeza de Dados (ETL).
- **16/12 a 22/12:** Análise Exploratória (EDA) e Engenharia de Atributos.
- **23/12 a 05/01:** Treinamento, Ajuste de Hiperparâmetros e Validação dos Modelos.
- **06/01:** Finalização da Escrita do Artigo e Preparação da Apresentação.
- **07/01:** Entrega Final.

## 6. Ferramentas

- Linguagem: Python 3.x
- Bibliotecas: Pandas, Scikit-Learn, Imbalanced-learn, Matplotlib/Seaborn.