

# **Trabalho Final da Disciplina de Probabilidade e Estatística**

## **Curso de Engenharia da Computação**

**Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)**

**Professor: Valberto Feitosa**

### **1. Objetivo**

Ao final da atividade, os alunos deverão ser capazes de **desenvolver uma solução completa de Machine Learning**, abrangendo desde a construção da base de dados até a disponibilização de um modelo treinado em uma aplicação web. Para isso, deverão:

1. **Construir uma base de dados** composta por imagens relacionadas a acidentes de trânsito, organizadas em três categorias (classes):
  - Acidente de trânsito grave
  - Acidente de trânsito moderado
  - Não é acidente
2. **Treinar um modelo de Machine Learning** utilizando o algoritmo **Naive Bayes**, a partir da base de dados construída.
3. **Salvar o modelo treinado**, permitindo sua reutilização posterior em um ambiente de aplicação real.
4. **Desenvolver uma aplicação web simples** (front-end e back-end) que possibilite ao usuário:
  - enviar novas imagens com características semelhantes às do conjunto de treinamento;
  - obter a **previsão da classe da imagem** (grave, moderado ou não é acidente) utilizando o **modelo salvo**.

---

### **2. Desenvolvimento da Atividade**

#### **2.1 Construção da Base de Dados**

Os alunos deverão coletar, organizar e rotular imagens de acidentes de trânsito, garantindo que cada imagem esteja corretamente associada a uma das três classes definidas. A base de dados deverá estar estruturada de forma adequada para o treinamento de modelos de Machine Learning.

---

#### **2.2 Parte 2: Treinamento do Modelo com Naive Bayes**

Os alunos deverão utilizar a linguagem **Python** e bibliotecas apropriadas, como o **scikit-learn**, para realizar as seguintes etapas:

1. Carregar a base de dados de imagens.
2. Realizar o pré-processamento necessário (extração de características, normalização, etc.).
3. Dividir os dados em conjuntos de **treinamento** e **teste**.
4. Treinar um modelo **Naive Bayes** utilizando o conjunto de treinamento.
5. Avaliar o desempenho do modelo no conjunto de teste, por meio de métricas como **acurácia**.

## **Salvamento do Modelo Treinado**

Após o treinamento e a avaliação, os alunos deverão **salvar o modelo treinado** para uso posterior, utilizando bibliotecas como:

- joblib, ou
- pickle.

O modelo deverá ser armazenado em um arquivo que possa ser carregado posteriormente pela aplicação web.

---

## **2.3 Parte 3: Desenvolvimento da Aplicação Web**

Os alunos deverão desenvolver uma **aplicação web simples** que permita a interação do usuário com o modelo treinado, possibilitando a inserção de novos dados e a obtenção de previsões.

### **2.3.1 Front-End**

O front-end da aplicação deverá:

- conter uma página HTML com um formulário para envio de dados (imagens);
- permitir que o usuário selecione ou envie uma imagem para análise;
- incluir um botão “**Prever**” para submeter os dados ao sistema.

**Observação:** Os alunos podem ser criativos e incluir informações adicionais sobre a imagem, como **coordenadas geográficas** ou **endereço do local do acidente**. Essas informações podem representar uma extensão importante do sistema, possibilitando, por exemplo, o **acionamento automático do serviço de socorro mais próximo**.

---

### **2.3.2 Back-End**

O back-end deverá ser desenvolvido utilizando **Flask**, **Django** ou outro framework web compatível, sendo responsável por:

- carregar o modelo previamente salvo;
- receber os dados enviados pelo formulário do front-end;
- aplicar o modelo **Naive Bayes** para realizar a previsão da classe da imagem;
- retornar o resultado da previsão ao front-end.

### **2.3.3 Exibição do Resultado**

A previsão gerada pelo modelo deverá ser exibida na própria página web, informando claramente ao usuário a **classe prevista** para a imagem enviada:

- Acidente de trânsito grave;
  - Acidente de trânsito moderado; ou
  - Não é acidente.
-

### **3. Exemplo de Ferramentas e Tecnologias**

- **Linguagem de Programação:** Python
  - **Bibliotecas de Machine Learning:** scikit-learn, pandas, numpy
  - **Bibliotecas para salvamento do modelo:** joblib ou pickle
  - **Back-End:** Flask ou Django
  - **Front-End:** HTML, CSS, JavaScript
  - **Comunicação Front-End / Back-End:** requisições HTTP do tipo POST (Fetch API ou AJAX)
- 

### **4. Fluxo da Aplicação**

1. O usuário acessa a página da aplicação web.
  2. Envia uma imagem por meio do formulário.
  3. Clica no botão “**Prever**”.
  4. O back-end processa a requisição, aplica o modelo treinado e realiza a previsão.
  5. O resultado da previsão (classe da imagem) é exibido na página web.
- 

### **5. Considerações Finais**

Este exercício permitirá aos alunos:

- aplicar o algoritmo **Naive Bayes** em um problema prático de Machine Learning;
- compreender o fluxo completo de um projeto de **Ciência de Dados e Machine Learning**, desde a construção da base até a implantação do modelo;
- desenvolver habilidades práticas em **desenvolvimento web**;
- integrar um modelo de Machine Learning a uma **aplicação web interativa**, simulando um cenário real de uso.

**Observação Final:** A equipe pode ser criativa e **adicionar funcionalidades extras à aplicação**, desde que mantenha os requisitos mínimos definidos nesta atividade.