# **Aluno: Higor Nogueira Souza / Sofware Product Data Analysis**

# **Documentação do Código**

## **Introdução**

Este documento fornece uma visão geral do código que implementa um modelo de aprendizado de máquina para análise de dados de filmes. O código é projetado para ser integrado em um projeto de software mais amplo

## **Objetivo**

O objetivo principal deste código é treinar um modelo de aprendizado de máquina para classificar filmes com base em uma variedade de recursos textuais, incluindo nome do filme, sinopse e gênero. O modelo deve ser treinado para prever categorias específicas para cada filme.

## **Componentes Principais**

### **1. Carregamento de Dados**

* A função **load\_data(train\_file, test\_file)** carrega dados de treinamento e teste a partir de arquivos CSV. Os dados carregados incluem informações sobre filmes.

### **2. Pré-processamento de Dados**

* A função **preprocess\_data(train\_data, test\_data)** realiza o pré-processamento dos dados. Isso envolve a combinação de várias colunas de texto e a tokenização das avaliações de filmes usando o **CountVectorizer**.

### **3. Treinamento do Modelo**

* A função **train\_model(model, train\_loader, criterion, optimizer, num\_epochs=10)** treina o modelo de aprendizado de máquina. O treinamento é realizado em várias épocas, calculando a perda e a precisão em cada época.

### **4. Criação de DataLoader**

* A função **create\_dataloader(X, y, batch\_size=32, shuffle=True)** cria um DataLoader para os dados de treinamento.

### **5. Salvar Precisão em um Arquivo CSV**

* A função **save\_accuracy\_to\_csv(filename, accuracy\_values)** salva os valores de precisão em um arquivo CSV.

## **Arquivos do Projeto**

* **train.csv** e **test.csv**: Arquivos CSV contendo os dados de treinamento e teste.
* **accuracy\_values.csv**: Arquivo CSV onde os valores de precisão são salvos.

## **Uso**

* Carregue os dados de treinamento e teste usando **load\_data(train\_file, test\_file)**.
* Pré-processe os dados usando **preprocess\_data(train\_data, test\_data)**.
* Crie um DataLoader para os dados de treinamento usando **create\_dataloader(X, y, batch\_size=32, shuffle=True)**.
* Defina a arquitetura do modelo e os hiperparâmetros.
* Treine o modelo usando **train\_model(model, train\_loader, criterion, optimizer, num\_epochs=10)**.
* Salve a precisão em um arquivo CSV com **save\_accuracy\_to\_csv(filename, accuracy\_values)**.