

Aula: Criando um Classificador de Frutas com Python e Tkinter

Olá, pessoal! Hoje vamos explorar um projeto muito interessante: um classificador de frutas utilizando Python, aprendizado de máquina e a biblioteca Tkinter para criar uma interface gráfica amigável.

1. Objetivo do Projeto

O objetivo deste projeto é treinar um modelo de machine learning para classificar palavras como "fruta" ou "não fruta" com base em um conjunto de treinamento armazenado em um arquivo de texto.

2. Estrutura do Código

Vamos entender cada parte do código:

Importação das Bibliotecas

```
import tkinter as tk
from tkinter import ttk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
import os
```

- **tkinter:** Responsável pela interface gráfica.
 - **sklearn:** Utilizado para criar o modelo de aprendizado de máquina.
 - **os:** Permite interagir com o sistema de arquivos.
-

Carregar Dados

```
def carregar_dados():
    palavras = []
    rotulos = []

    if os.path.exists('caminho_do_arquivo.txt'):
        with open('caminho_do_arquivo.txt', 'r') as f:
            for linha in f:
                try:
                    palavra, rotulo = linha.strip().split(',')
                    palavras.append(palavra)
                    rotulos.append(int(rotulo))
                except ValueError:
                    print(f"Erro ao processar a linha: {linha.strip()}")

    return palavras, rotulos
```

- Lê as palavras e seus respectivos rótulos (0 para "não é uma fruta", 1 para "é uma fruta") de um arquivo de texto.
 - Caso ocorra um erro no formato da linha, ele exibe uma mensagem de erro.
-

Treinar Modelo

```
def treinar_modelo(palavras, rotulos):  
    modelo = make_pipeline(CountVectorizer(), MultinomialNB())  
    modelo.fit(palavras, rotulos)  
    return modelo
```

- **CountVectorizer** transforma as palavras em valores numéricos.
 - **MultinomialNB** é um classificador probabilístico baseado no Teorema de Bayes.
-

Adicionar uma Nova Palavra

```
def adicionar_palavra():  
    global modelo, palavras, rotulos  
    nova_palavra = entrada_palavra.get()  
    novo_rotulo = var_rotulo.get()  
  
    if nova_palavra and (novo_rotulo == 0 or novo_rotulo == 1):  
        if nova_palavra not in palavras:  
            palavras.append(nova_palavra.lower())  
            rotulos.append(novo_rotulo)  
            salvar_dados(palavras, rotulos)  
            modelo = treinar_modelo(palavras, rotulos)  
            lbl_resultado['text'] = f"A palavra '{nova_palavra}' foi adicionada  
e o modelo foi atualizado."  
        else:  
            lbl_resultado['text'] = f"A palavra '{nova_palavra}' já está no  
conjunto de treinamento."
```

- Obtém a palavra digitada pelo usuário e o rótulo selecionado.
 - Adiciona a palavra ao conjunto de treinamento caso ainda não exista.
 - Salva os dados e atualiza o modelo.
-

Verificar uma Palavra

```
def verificar_palavra():  
    global modelo, palavras, rotulos  
    palavras, rotulos = carregar_dados()  
    modelo = treinar_modelo(palavras, rotulos)  
  
    consulta_palavra = entrada_palavra.get().lower()  
    if consulta_palavra in palavras:  
        previsao = modelo.predict([consulta_palavra])  
        lbl_resultado['text'] = f"Previsão para '{consulta_palavra}': {'É uma  
Fruta' if previsao[0] == 1 else 'Não é uma fruta'}"  
    else:  
        lbl_resultado['text'] = f"Palavra '{consulta_palavra}' não encontrada."
```

- Verifica se a palavra está na base de dados e faz uma previsão usando o modelo treinado.
-

3. Criando a Interface Gráfica (GUI)

```
janela = tk.Tk()  
janela.geometry("900x300")  
janela.title("Classificador de Frutas")
```

- Criamos a janela principal do aplicativo.

A interface conta com:

- **Um campo de entrada para a palavra**
 - **Botões para selecionar se é ou não uma fruta**
 - **Botões para adicionar e verificar palavras**
 - **Um rótulo para exibir o resultado**
-

Conclusão

Este projeto é um ótimo exemplo de como podemos combinar aprendizado de máquina e interfaces gráficas para criar aplicações interativas. Você pode expandi-lo adicionando mais categorias ou integrando outras técnicas de machine learning.

Se gostou do projeto, compartilhe e marque seus amigos que também curtem programação! 🚀💡
