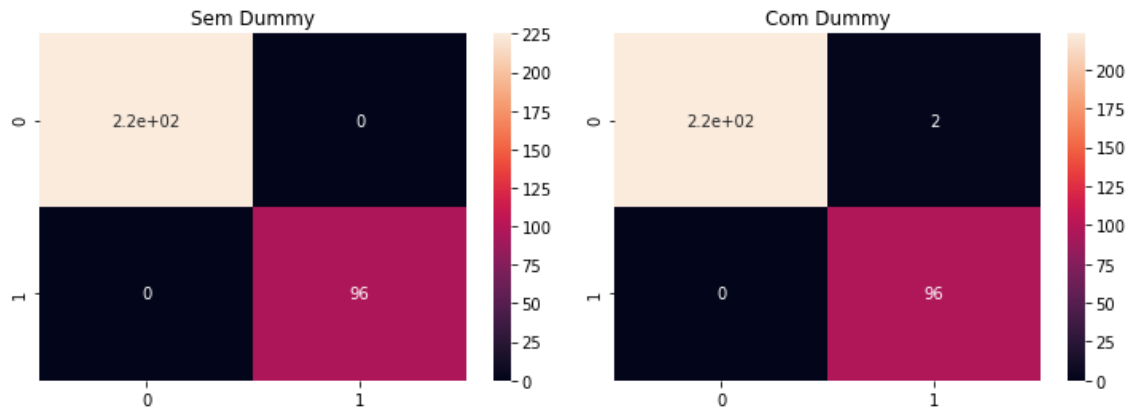
```
# -*- coding: utf-8 -*-
"""
@author: Laura
"""

import pandas as pd
import numpy as np
from seaborn import heatmap
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt

base_dados = pd.read_csv('bd_tarefa2.CSV', sep=';')
estatisticas = base_dados.describe()
base_dados.columns = ['CT', 'CU', 'LT', 'TC', 'TS']

classe = base_dados.iloc[:,4] #pego todas as linhas da coluna 4 - trustworth
vetor = np.array(base_dados.iloc[:,0:4]) #todas as linhas das colunas 0 até 4

#converter dados textuais em numericos para interpretação do programa
le = LabelEncoder()
vetor[:, 0] = le.fit_transform(vetor[:,0])
vetor[:, 1] = le.fit_transform(vetor[:,1])
vetor[:, 2] = le.fit_transform(vetor[:,2])
vetor[:, 3] = le.fit_transform(vetor[:,3])

#criacao dos dummies a partir do arquivo
db_dummy = pd.get_dummies(base_dados)
previsores_dummy = np.array(db_dummy.iloc[:,0:16])

#classificando SEM dummy
primeira = KNeighborsClassifier(n_neighbors=20)
primeira.fit(vetor, classe)
resultado1 = primeira.predict(vetor)

#classificando COM dummy
segunda = KNeighborsClassifier(n_neighbors=20)
segunda.fit(previsores_dummy, classe)
resultado2 = segunda.predict(previsores_dummy)

matriz1 = confusion_matrix(classe, resultado1)
matriz2 = confusion_matrix(classe, resultado2)

#gerar as matrizes de maneira gráfica e visual
m1 = pd.DataFrame(matriz1, index = [i for i in "01"], columns = [i for i in "01"])
m2 = pd.DataFrame(matriz2, index = [i for i in "01"], columns = [i for i in "01"])
plt.title("Sem Dummy")
heatmap(m1, annot=True)
plt.show()
plt.title("Com Dummy")
heatmap(m2, annot=True)
plt.show()
```



Conclusões:

Para esta base de dados, a utilização do *Dummy* acabou não favorecendo tanto a divisão dos dados. Como o número de amostras é alto (321 com 4 colunas) o número de vizinhos (“*n_neighbors*”) definidos foi de 20.

Obs: Alguns testes foram feitos, com valores menores e maiores para compreensão do algoritmo, e, considerando 100 vizinhos os resultados tiveram mais erros (76 sem o *dummy* e 96 com), já considerando 5 vizinhos não houveram erros em ambas amostras. Então para poder avaliar melhor a influencia da *dummy* foi utilizado o valor de 20 vizinhos.

Nos resultados obtidos sem o uso da base *dummy* o algoritmo não obteve erros de previsões (nos pontos em que o algoritmo previu que seriam 0, 225 eram de fato zero – acerto de 100%, e das previsões de 1 também 100% de acerto).

Já com o uso da base dummy, das previsões onde seriam 0 houveram dois erros (dois pontos previstos como 0 são na realidade 1).

Assim, para a base de dados utilizada, o *dummy* não auxiliou na previsão dos dados.