**CSE422\_Lab04(only code):**

from google.colab import files

load\_data = files.upload()

import numpy as np

import pandas as pd

import io

import sklearn

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import MinMaxScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

#loading data

dataset = pd.read\_csv(io.BytesIO(load\_data['mushroom edibility classification dataset.csv']))

dataset.shape

# missing values

dataset.drop(dataset.columns[dataset.columns.str.contains('unnamed',case = False)],axis = 1, inplace = True)

dataset.head()

print(dataset.isnull().sum())

dataset=dataset.dropna(how = 'any', axis = 0)

dataset.shape

dataset = dataset.dropna(axis = 0, subset = ['cap-shape', 'cap-color'])

print(dataset.isnull().sum())

dataset.shape

print(dataset.info())

#encoding categorical features

print(dataset['class'].unique())

print(dataset['bruises'].unique())

enc = LabelEncoder()

dataset['class'] = enc.fit\_transform(dataset['class'])

dataset['bruises'] = enc.fit\_transform(dataset['bruises'])

dataset[['class', 'bruises']].head()

#scaling data with minmaxscaler

scaler = MinMaxScaler()

scaler.fit(dataset)

dataset\_train\_scaled = scaler.transform(dataset)

print('per-feature minimum before scaling:\n{}'.format(dataset.min(axis = 0)))

print('per-feature maximum before scaling:\n{}'.format(dataset.max(axis = 0)))

print('per-feature minimum after scaling:\n{}'.format(dataset\_train\_scaled.min(axis = 0)))

print('per-feature maximum after scaling:\n{}'.format(dataset\_train\_scaled.max(axis = 0)))

#splitting the dataset into features and labels

features = dataset[['cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor', 'stalk-shape', 'stalk-root', 'stalk-surface-above-ring', 'stalk-surface-below-ring', 'stalk-color-above-ring', 'stalk-color-below-ring', 'veil-type', 'veil-color', 'ring-number', 'ring-type', 'spore-print-color', 'population', 'habitat']]

label = dataset[['class']]

stratified = pd.DataFrame(label)

xTrain, xTest, yTrain, yTest = train\_test\_split(features, label, test\_size = 0.25, stratify = stratified, random\_state = 0)

knn = KNeighborsClassifier()

knn.fit(xTrain, yTrain)

print("Test set accuracy: {:.2f}".format(knn.score(xTest, yTest)))

**Code with outputs:**

{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"name": "Untitled0.ipynb",

"provenance": [],

"collapsed\_sections": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "code",

"metadata": {

"colab": {

"resources": {

"http://localhost:8080/nbextensions/google.colab/files.js": {

"data": "",

"ok": true,

"headers": [

[

"content-type",

"application/javascript"

]

],

"status": 200,

"status\_text": ""

}

},

"base\_uri": "https://localhost:8080/",

"height": 73

},

"id": "74YVilcoMgLg",

"outputId": "333eb119-a10c-4ac4-e7be-9c7bd9783992"

},

"source": [

" from google.colab import files\n",

" load\_data = files.upload()"

],

"execution\_count": 1,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/html": [

"\n",

" <input type=\"file\" id=\"files-41b87876-bb0e-403c-90aa-37cdbe7c97b6\" name=\"files[]\" multiple disabled\n",

" style=\"border:none\" />\n",

" <output id=\"result-41b87876-bb0e-403c-90aa-37cdbe7c97b6\">\n",

" Upload widget is only available when the cell has been executed in the\n",

" current browser session. Please rerun this cell to enable.\n",

" </output>\n",

" <script src=\"/nbextensions/google.colab/files.js\"></script> "

],

"text/plain": [

"<IPython.core.display.HTML object>"

]

},

"metadata": {}

},

{

"output\_type": "stream",

"text": [

"Saving mushroom edibility classification dataset.csv to mushroom edibility classification dataset.csv\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "SrTh3IhsMqYS",

"outputId": "81c7a507-eaeb-43d1-9877-fbe5d810f067"

},

"source": [

" import numpy as np\n",

" import pandas as pd\n",

" import io\n",

" import sklearn\n",

" from sklearn.preprocessing import LabelEncoder\n",

" from sklearn.preprocessing import MinMaxScaler\n",

" from sklearn.model\_selection import train\_test\_split\n",

" from sklearn.neighbors import KNeighborsClassifier\n",

" \n",

" #loading data \n",

" dataset = pd.read\_csv(io.BytesIO(load\_data['mushroom edibility classification dataset.csv']))\n",

" dataset.shape"

],

"execution\_count": 5,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(3124, 20)"

]

},

"metadata": {},

"execution\_count": 5

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 338

},

"id": "xBvegGw9N2ys",

"outputId": "13a19f0a-eb18-4813-d50c-bd8790a26238"

},

"source": [

" # missing values\n",

" dataset.drop(dataset.columns[dataset.columns.str.contains('unnamed',case = False)],axis = 1, inplace = True)\n",

" dataset.head()\n"

],

"execution\_count": 6,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>class</th>\n",

" <th>cap-shape</th>\n",

" <th>cap-surface</th>\n",

" <th>cap-color</th>\n",

" <th>bruises</th>\n",

" <th>odor</th>\n",

" <th>stalk-shape</th>\n",

" <th>stalk-root</th>\n",

" <th>stalk-surface-above-ring</th>\n",

" <th>stalk-surface-below-ring</th>\n",

" <th>stalk-color-above-ring</th>\n",

" <th>stalk-color-below-ring</th>\n",

" <th>veil-type</th>\n",

" <th>veil-color</th>\n",

" <th>ring-number</th>\n",

" <th>ring-type</th>\n",

" <th>spore-print-color</th>\n",

" <th>population</th>\n",

" <th>habitat</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>poisonous</td>\n",

" <td>5.0</td>\n",

" <td>2</td>\n",

" <td>4.0</td>\n",

" <td>bruises is there</td>\n",

" <td>6</td>\n",

" <td>0</td>\n",

" <td>3</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>7</td>\n",

" <td>7</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>4</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>5</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>edible</td>\n",

" <td>5.0</td>\n",

" <td>2</td>\n",

" <td>9.0</td>\n",

" <td>bruises is there</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>7</td>\n",

" <td>7</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>4</td>\n",

" <td>3</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>edible</td>\n",

" <td>0.0</td>\n",

" <td>2</td>\n",

" <td>8.0</td>\n",

" <td>bruises is there</td>\n",

" <td>3</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>7</td>\n",

" <td>7</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>4</td>\n",

" <td>3</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>poisonous</td>\n",

" <td>5.0</td>\n",

" <td>3</td>\n",

" <td>8.0</td>\n",

" <td>bruises is there</td>\n",

" <td>6</td>\n",

" <td>0</td>\n",

" <td>3</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>7</td>\n",

" <td>7</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>4</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>5</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>edible</td>\n",

" <td>5.0</td>\n",

" <td>2</td>\n",

" <td>3.0</td>\n",

" <td>no bruises</td>\n",

" <td>5</td>\n",

" <td>1</td>\n",

" <td>3</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>7</td>\n",

" <td>7</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>3</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" class cap-shape cap-surface ... spore-print-color population habitat\n",

"0 poisonous 5.0 2 ... 2 3 5\n",

"1 edible 5.0 2 ... 3 2 1\n",

"2 edible 0.0 2 ... 3 2 3\n",

"3 poisonous 5.0 3 ... 2 3 5\n",

"4 edible 5.0 2 ... 3 0 1\n",

"\n",

"[5 rows x 19 columns]"

]

},

"metadata": {},

"execution\_count": 6

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "15uboPVoap58",

"outputId": "39c64e4f-adec-45dc-ede7-c7a724a11c62"

},

"source": [

"print(dataset.isnull().sum())\n",

"dataset=dataset.dropna(how = 'any', axis = 0)\n",

"dataset.shape\n",

"dataset = dataset.dropna(axis = 0, subset = ['cap-shape', 'cap-color'])\n",

"print(dataset.isnull().sum())\n",

"dataset.shape\n",

"print('\*\*\*\*\*\*\*\*\*\*')\n",

"print(dataset.info())"

],

"execution\_count": 7,

"outputs": [

{

"output\_type": "stream",

"text": [

"class 0\n",

"cap-shape 8\n",

"cap-surface 0\n",

"cap-color 8\n",

"bruises 0\n",

"odor 0\n",

"stalk-shape 0\n",

"stalk-root 0\n",

"stalk-surface-above-ring 0\n",

"stalk-surface-below-ring 0\n",

"stalk-color-above-ring 0\n",

"stalk-color-below-ring 0\n",

"veil-type 0\n",

"veil-color 0\n",

"ring-number 0\n",

"ring-type 0\n",

"spore-print-color 0\n",

"population 0\n",

"habitat 0\n",

"dtype: int64\n",

"class 0\n",

"cap-shape 0\n",

"cap-surface 0\n",

"cap-color 0\n",

"bruises 0\n",

"odor 0\n",

"stalk-shape 0\n",

"stalk-root 0\n",

"stalk-surface-above-ring 0\n",

"stalk-surface-below-ring 0\n",

"stalk-color-above-ring 0\n",

"stalk-color-below-ring 0\n",

"veil-type 0\n",

"veil-color 0\n",

"ring-number 0\n",

"ring-type 0\n",

"spore-print-color 0\n",

"population 0\n",

"habitat 0\n",

"dtype: int64\n",

"\*\*\*\*\*\*\*\*\*\*\n",

"<class 'pandas.core.frame.DataFrame'>\n",

"Int64Index: 3108 entries, 0 to 3123\n",

"Data columns (total 19 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 class 3108 non-null object \n",

" 1 cap-shape 3108 non-null float64\n",

" 2 cap-surface 3108 non-null int64 \n",

" 3 cap-color 3108 non-null float64\n",

" 4 bruises 3108 non-null object \n",

" 5 odor 3108 non-null int64 \n",

" 6 stalk-shape 3108 non-null int64 \n",

" 7 stalk-root 3108 non-null int64 \n",

" 8 stalk-surface-above-ring 3108 non-null int64 \n",

" 9 stalk-surface-below-ring 3108 non-null int64 \n",

" 10 stalk-color-above-ring 3108 non-null int64 \n",

" 11 stalk-color-below-ring 3108 non-null int64 \n",

" 12 veil-type 3108 non-null int64 \n",

" 13 veil-color 3108 non-null int64 \n",

" 14 ring-number 3108 non-null int64 \n",

" 15 ring-type 3108 non-null int64 \n",

" 16 spore-print-color 3108 non-null int64 \n",

" 17 population 3108 non-null int64 \n",

" 18 habitat 3108 non-null int64 \n",

"dtypes: float64(2), int64(15), object(2)\n",

"memory usage: 485.6+ KB\n",

"None\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Y9Un6QhqeAJR",

"outputId": "8745b096-b3e7-4ed6-b141-4d2a2df9bdb8"

},

"source": [

" #encoding categorical features\n",

" print(dataset['class'].unique())\n",

" print(dataset['bruises'].unique())"

],

"execution\_count": 8,

"outputs": [

{

"output\_type": "stream",

"text": [

"['poisonous' 'edible']\n",

"['bruises is there' 'no bruises']\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "GXzsePAdtvcy",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "7e767f64-1873-4460-8398-30ec43283171"

},

"source": [

"from sklearn.preprocessing import LabelEncoder\n",

"from sklearn.preprocessing import MinMaxScaler\n",

"\n",

"enc = LabelEncoder()\n",

"dataset['class'] = enc.fit\_transform(dataset['class'])\n",

"dataset['bruises'] = enc.fit\_transform(dataset['bruises'])\n",

"dataset[['class', 'bruises']].head()\n",

"\n",

"scaler = MinMaxScaler()\n",

"scaler.fit(dataset)\n",

"dataset\_train\_scaled = scaler.transform(dataset)\n",

"print('per-feature minimum before scaling:\\n{}'.format(dataset.min(axis = 0)))\n",

"print('per-feature maximum before scaling:\\n{}'.format(dataset.max(axis = 0)))\n",

"print('per-feature minimum after scaling:\\n{}'.format(dataset\_train\_scaled.min(axis = 0)))\n",

"print('per-feature maximum after scaling:\\n{}'.format(dataset\_train\_scaled.max(axis = 0)))\n",

"\n"

],

"execution\_count": 9,

"outputs": [

{

"output\_type": "stream",

"text": [

"per-feature minimum before scaling:\n",

"class 0.0\n",

"cap-shape 0.0\n",

"cap-surface 0.0\n",

"cap-color 2.0\n",

"bruises 0.0\n",

"odor 0.0\n",

"stalk-shape 0.0\n",

"stalk-root 1.0\n",

"stalk-surface-above-ring 0.0\n",

"stalk-surface-below-ring 0.0\n",

"stalk-color-above-ring 0.0\n",

"stalk-color-below-ring 0.0\n",

"veil-type 0.0\n",

"veil-color 2.0\n",

"ring-number 1.0\n",

"ring-type 0.0\n",

"spore-print-color 1.0\n",

"population 0.0\n",

"habitat 0.0\n",

"dtype: float64\n",

"per-feature maximum before scaling:\n",

"class 1.0\n",

"cap-shape 5.0\n",

"cap-surface 3.0\n",

"cap-color 9.0\n",

"bruises 1.0\n",

"odor 6.0\n",

"stalk-shape 1.0\n",

"stalk-root 4.0\n",

"stalk-surface-above-ring 2.0\n",

"stalk-surface-below-ring 3.0\n",

"stalk-color-above-ring 7.0\n",

"stalk-color-below-ring 7.0\n",

"veil-type 0.0\n",

"veil-color 2.0\n",

"ring-number 1.0\n",

"ring-type 4.0\n",

"spore-print-color 6.0\n",

"population 5.0\n",

"habitat 5.0\n",

"dtype: float64\n",

"per-feature minimum after scaling:\n",

"[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n",

"per-feature maximum after scaling:\n",

"[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 1. 1. 1. 1.]\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "Lo\_k\_F6VZ1JX",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "cd6b2b8a-e427-49be-f3c3-fd2c76dbb5b1"

},

"source": [

"#spliting the dataset into features and labels\n",

"features = dataset[['cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor', 'stalk-shape', 'stalk-root', 'stalk-surface-above-ring', 'stalk-surface-below-ring', 'stalk-color-above-ring', 'stalk-color-below-ring', 'veil-type', 'veil-color', 'ring-number', 'ring-type', 'spore-print-color', 'population', 'habitat']]\n",

"label = dataset[['class']]\n",

"stratified = pd.DataFrame(label)\n",

"xTrain, xTest, yTrain, yTest = train\_test\_split(features, label, test\_size = 0.25, stratify = stratified, random\_state = 0)\n",

"\n",

"knn = KNeighborsClassifier()\n",

"knn.fit(xTrain, yTrain)\n",

"print(\"Test set accuracy: {:.2f}\".format(knn.score(xTest, yTest)))"

],

"execution\_count": 10,

"outputs": [

{

"output\_type": "stream",

"text": [

"Test set accuracy: 1.00\n"

],

"name": "stdout"

},

{

"output\_type": "stream",

"text": [

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:8: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().\n",

" \n"

],

"name": "stderr"

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "nmRRRpz9973B"

},

"source": [

""

],

"execution\_count": null,

"outputs": []

}

]

}