DATA OUTLINE-

1.Different types of fruits with the fruit\_label

2.Mass of the each fruit

3.Width of each fruit

4.Height of each fruit

5.Color\_Score of each fruit.

CODE-

{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"kernelspec": {

"display\_name": "Python 3",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.7.1"

},

"colab": {

"name": "Problem Statement 1.ipynb",

"provenance": [],

"collapsed\_sections": [],

"include\_colab\_link": true

}

},

"cells": [

{

"cell\_type": "markdown",

"metadata": {

"id": "view-in-github",

"colab\_type": "text"

},

"source": [

"<a href=\"https://colab.research.google.com/github/Sanjeeth-S/Curneu-Assessment/blob/main/Problem\_Statement\_1.ipynb\" target=\"\_parent\"><img src=\"https://colab.research.google.com/assets/colab-badge.svg\" alt=\"Open In Colab\"/></a>"

]

},

{

"cell\_type": "code",

"metadata": {

"id": "hwbPhpdoVVpw"

},

"source": [

"import pandas as pd\r\n",

"import matplotlib.pyplot as plt\r\n",

"import numpy as np"

],

"execution\_count": 53,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "fsTrNU8gW-\_\_"

},

"source": [

""

],

"execution\_count": 4,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "DjnalbecVVpy"

},

"source": [

"datae = pd.read\_excel(\"fruits.xlsx\")"

],

"execution\_count": 54,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

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

},

"id": "Dj6Z9XzeYFyu",

"outputId": "17106a4f-47f3-4e08-e23d-54eafa08eae1"

},

"source": [

"datae.info()"

],

"execution\_count": 55,

"outputs": [

{

"output\_type": "stream",

"text": [

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

"RangeIndex: 59 entries, 0 to 58\n",

"Data columns (total 6 columns):\n",

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

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

" 0 fruit\_label 59 non-null int64 \n",

" 1 fruit\_name 59 non-null object \n",

" 2 mass 59 non-null int64 \n",

" 3 width 59 non-null float64\n",

" 4 height 59 non-null float64\n",

" 5 color\_score 59 non-null float64\n",

"dtypes: float64(3), int64(2), object(1)\n",

"memory usage: 2.9+ KB\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "8UzvyxsvW\_hR"

},

"source": [

"from sklearn.preprocessing import LabelEncoder\r\n",

"k = LabelEncoder()\r\n",

"datae['fruit\_name'] = k.fit\_transform(datae['fruit\_name'])"

],

"execution\_count": 56,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

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

"height": 294

},

"id": "BscOdtRKXL8v",

"outputId": "6a106754-857c-4f46-9d16-f9d04afe98d3"

},

"source": [

"datae.describe()"

],

"execution\_count": 57,

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

" <td>59.000000</td>\n",

" <td>59.000000</td>\n",

" <td>59.000000</td>\n",

" <td>59.000000</td>\n",

" <td>59.000000</td>\n",

" <td>59.000000</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>2.542373</td>\n",

" <td>1.406780</td>\n",

" <td>163.118644</td>\n",

" <td>7.105085</td>\n",

" <td>7.693220</td>\n",

" <td>0.762881</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>1.208048</td>\n",

" <td>1.247323</td>\n",

" <td>55.018832</td>\n",

" <td>0.816938</td>\n",

" <td>1.361017</td>\n",

" <td>0.076857</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>1.000000</td>\n",

" <td>0.000000</td>\n",

" <td>76.000000</td>\n",

" <td>5.800000</td>\n",

" <td>4.000000</td>\n",

" <td>0.550000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>1.000000</td>\n",

" <td>0.000000</td>\n",

" <td>140.000000</td>\n",

" <td>6.600000</td>\n",

" <td>7.200000</td>\n",

" <td>0.720000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>3.000000</td>\n",

" <td>1.000000</td>\n",

" <td>158.000000</td>\n",

" <td>7.200000</td>\n",

" <td>7.600000</td>\n",

" <td>0.750000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>4.000000</td>\n",

" <td>3.000000</td>\n",

" <td>177.000000</td>\n",

" <td>7.500000</td>\n",

" <td>8.200000</td>\n",

" <td>0.810000</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>4.000000</td>\n",

" <td>3.000000</td>\n",

" <td>362.000000</td>\n",

" <td>9.600000</td>\n",

" <td>10.500000</td>\n",

" <td>0.930000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" fruit\_label fruit\_name mass width height color\_score\n",

"count 59.000000 59.000000 59.000000 59.000000 59.000000 59.000000\n",

"mean 2.542373 1.406780 163.118644 7.105085 7.693220 0.762881\n",

"std 1.208048 1.247323 55.018832 0.816938 1.361017 0.076857\n",

"min 1.000000 0.000000 76.000000 5.800000 4.000000 0.550000\n",

"25% 1.000000 0.000000 140.000000 6.600000 7.200000 0.720000\n",

"50% 3.000000 1.000000 158.000000 7.200000 7.600000 0.750000\n",

"75% 4.000000 3.000000 177.000000 7.500000 8.200000 0.810000\n",

"max 4.000000 3.000000 362.000000 9.600000 10.500000 0.930000"

]

},

"metadata": {

"tags": []

},

"execution\_count": 57

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

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

"height": 283

},

"id": "OE-xbkHBVVpz",

"outputId": "77a10b19-2b15-47b3-d28d-008ff5100d32"

},

"source": [

"#df=datae.iloc[:,2:6]\n",

"#tar=data.iloc[:,1]\n",

"X=datae.color\_score\n",

"y=datae.fruit\_label\n",

"plt.scatter(X,y)"

],

"execution\_count": 58,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.collections.PathCollection at 0x7f3190b4d610>"

]

},

"metadata": {

"tags": []

},

"execution\_count": 58

},

{

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

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "aKh0wd9BXtky"

},

"source": [

"X = datae.drop('fruit\_label', axis = 1)\r\n",

"y = datae['fruit\_label']"

],

"execution\_count": 59,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "sJyTu9ueVVpz"

},

"source": [

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

"X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)"

],

"execution\_count": 60,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "BLtC64DtVVp0"

},

"source": [

"from sklearn.preprocessing import StandardScaler\n",

"scaler = StandardScaler()\n",

"scaler.fit(X\_train)\n",

"\n",

"X\_train = scaler.transform(X\_train)\n",

"X\_test = scaler.transform(X\_test)"

],

"execution\_count": 61,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

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

},

"id": "oDkwoN0XaEfK",

"outputId": "71278631-8433-4760-fc39-5f01a70708db"

},

"source": [

"datae"

],

"execution\_count": 52,

"outputs": [

{

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

"data": {

"text/plain": [

"array([0, 0, 0, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",

" 0, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1,\n",

" 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])"

]

},

"metadata": {

"tags": []

},

"execution\_count": 52

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

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

},

"id": "2wfIPklxVVp0",

"outputId": "765d6ab8-fb79-497e-981c-959ba6b4ca47"

},

"source": [

"from sklearn.neighbors import KNeighborsClassifier\n",

"classifier = KNeighborsClassifier(n\_neighbors=5)\n",

"classifier.fit(X\_train, y\_train)"

],

"execution\_count": 62,

"outputs": [

{

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

"data": {

"text/plain": [

"KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',\n",

" metric\_params=None, n\_jobs=None, n\_neighbors=5, p=2,\n",

" weights='uniform')"

]

},

"metadata": {

"tags": []

},

"execution\_count": 62

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "QU7fumkfVVp1"

},

"source": [

"y\_pred = classifier.predict(X\_test)"

],

"execution\_count": 63,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

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

},

"id": "Xw75cFH-VVp1",

"outputId": "7e81cc86-acaf-414e-80c4-ab26b8553784"

},

"source": [

"from sklearn.metrics import classification\_report, confusion\_matrix\n",

"print(confusion\_matrix(y\_test, y\_pred))\n",

"print(classification\_report(y\_test, y\_pred))"

],

"execution\_count": 64,

"outputs": [

{

"output\_type": "stream",

"text": [

"[[4 0 0 0]\n",

" [0 1 0 0]\n",

" [0 0 4 0]\n",

" [0 0 0 3]]\n",

" precision recall f1-score support\n",

"\n",

" 1 1.00 1.00 1.00 4\n",

" 2 1.00 1.00 1.00 1\n",

" 3 1.00 1.00 1.00 4\n",

" 4 1.00 1.00 1.00 3\n",

"\n",

" accuracy 1.00 12\n",

" macro avg 1.00 1.00 1.00 12\n",

"weighted avg 1.00 1.00 1.00 12\n",

"\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "\_MID43SPVVp1"

},

"source": [

"error = []\n",

"\n",

"# Calculating error for K values between 1 and 40\n",

"for i in range(1, 40):\n",

" knn = KNeighborsClassifier(n\_neighbors=i)\n",

" knn.fit(X\_train, y\_train)\n",

" pred\_i = knn.predict(X\_test)\n",

" error.append(np.mean(pred\_i != y\_test))"

],

"execution\_count": 65,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

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

"height": 421

},

"id": "aZNsSOB3VVp2",

"outputId": "477b28f8-c6e8-418a-e668-333aa33c9d5a"

},

"source": [

"import matplotlib.pyplot as plt\n",

"plt.figure(figsize=(12, 6))\n",

"plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',\n",

" markerfacecolor='blue', markersize=10)\n",

"plt.title('Error Rate K Value')\n",

"plt.xlabel('K Value')\n",

"plt.ylabel('Mean Error')"

],

"execution\_count": 66,

"outputs": [

{

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

"data": {

"text/plain": [

"Text(0, 0.5, 'Mean Error')"

]

},

"metadata": {

"tags": []

},

"execution\_count": 66

},

{

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

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x432 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "aayqAId7VVp2"

},

"source": [

""

],

"execution\_count": 66,

"outputs": []

}

]

}