```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "name": " IRIS_KNN.ipynb",
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "metadata": {
        "id": "LTitY9WKOn8q"
      },
      "source": [
        "import pandas as pd\n",
        "import numpy as np\n",
        "import sklearn\n",
        "import seaborn as sns\n",
        "import matplotlib.pyplot as plt\n",
        "from sklearn.preprocessing import StandardScaler\n",
        "from sklearn import metrics\n",
        "from sklearn import model_selection\n",
        "from sklearn.metrics import classification_report\n",
        "from sklearn.metrics import confusion_matrix\n",
        "from sklearn.metrics import accuracy_score\n",
        "from sklearn.neighbors import KNeighborsClassifier\n",
        "from sklearn.metrics import *\n",
        "from sklearn.model_selection import *\n",
        "from sklearn.model_selection import train_test_split\n",
        "from sklearn.neighbors import KNeighborsClassifier\n"
      ],
      "execution_count": 66,
      "outputs": []
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "MpQnTcpdPBp7"
      },
      "source": [
```

```
        "open_dataset = \"/content/iris.csv\"\n",
        "names = ['sepal-length', 'sepal-width', 'petal-length',
'petal-width', 'class']\n",
        "dataset = pandas.read_csv(open_dataset, names=names)"
      ],
      "execution_count": 67,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "VcSDdsU-ksm7"
      },
      "source": [
        "# New Section"
      ]
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "bNWDOJvEPIBj",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "b7967094-9b34-4585-f431-5420a3219961"
      },
      "source": [
        "print(dataset.shape)"
      ],
      "execution_count": 68,
      "outputs": [
        {
          "output_type": "stream",
          "text": [
            "(151, 5)\n"
          ],
          "name": "stdout"
        }
      ]
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "u1eiUcVGPLcp",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "7b48dc4a-e6f5-4fe3-f240-5170afa165b0"
      },
      "source": [
```

```
        "print(dataset.head(20))"
      ],
      "execution_count": 69,
      "outputs": [
        {
          "output_type": "stream",
          "text": [
            "     sepal-length   sepal-width   petal-length   petal-
width        class\n",
            "Id  SepalLengthCm  SepalWidthCm  PetalLengthCm
PetalWidthCm      Species\n",
            "1             5.1           3.5            1.4
0.2  Iris-setosa\n",
            "2             4.9           3.0            1.4
0.2  Iris-setosa\n",
            "3             4.7           3.2            1.3
0.2  Iris-setosa\n",
            "4             4.6           3.1            1.5
0.2  Iris-setosa\n",
            "5             5.0           3.6            1.4
0.2  Iris-setosa\n",
            "6             5.4           3.9            1.7
0.4  Iris-setosa\n",
            "7             4.6           3.4            1.4
0.3  Iris-setosa\n",
            "8             5.0           3.4            1.5
0.2  Iris-setosa\n",
            "9             4.4           2.9            1.4
0.2  Iris-setosa\n",
            "10            4.9           3.1            1.5
0.1  Iris-setosa\n",
            "11            5.4           3.7            1.5
0.2  Iris-setosa\n",
            "12            4.8           3.4            1.6
0.2  Iris-setosa\n",
            "13            4.8           3.0            1.4
0.1  Iris-setosa\n",
            "14            4.3           3.0            1.1
0.1  Iris-setosa\n",
            "15            5.8           4.0            1.2
0.2  Iris-setosa\n",
            "16            5.7           4.4            1.5
0.4  Iris-setosa\n",
            "17            5.4           3.9            1.3
0.4  Iris-setosa\n",
            "18            5.1           3.5            1.4
0.3  Iris-setosa\n",
            "19            5.7           3.8            1.7
0.3  Iris-setosa\n"
          ],
```

```
          "name": "stdout"
        }
      ]
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "c_5FtC6JPLfN",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "95f9090c-4975-4796-bb89-ba1e2bbe487e"
      },
      "source": [
        "print(dataset.describe())"
      ],
      "execution_count": 70,
      "outputs": [
        {
          "output_type": "stream",
          "text": [
            "       sepal-length sepal-width petal-length petal-width
class\n",
            "count           151         151          151         151
151\n",
            "unique           36          24           44          23
4\n",
            "top             5.0         3.0          1.5         0.2
Iris-versicolor\n",
            "freq             10          26           14          28
50\n"
          ],
          "name": "stdout"
        }
      ]
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "CYPhsYEZPtNk",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "146f8d1c-81af-4d6c-a3a8-8e87132be07a"
      },
      "source": [
        "print(dataset.groupby('class').size())"
      ],
      "execution_count": 71,
      "outputs": [
```

```
        {
          "output_type": "stream",
          "text": [
            "class\n",
            "Iris-setosa        50\n",
            "Iris-versicolor    50\n",
            "Iris-virginica     50\n",
            "Species             1\n",
            "dtype: int64\n"
          ],
          "name": "stdout"
        }
      ]
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "3o7TO2UYQDKI"
      },
      "source": [
        "array = dataset.values\n",
        "X = array[:,0:4]\n",
        "Y = array[:,4]\n",
        "validation_size = 0.20\n",
        "seed = 7\n",
        "X_train, X_validation, Y_train,  Y_validation =
model_selection.train_test_split(X, Y, test_size=validation_size,
random_state=seed)\n",
        "\n",
        "seed = 7\n",
        "scoring = 'accuracy'"
      ],
      "execution_count": 72,
      "outputs": []
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "AhiyphKHQOOi",
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 1000
        },
        "outputId": "c2eddd47-0373-4362-d37b-5d4ab9ca3f62"
      },
      "source": [
        "kfold = model_selection.KFold(n_splits=10, random_state=seed)
\n",
        "cv_results =
model_selection.cross_val_score(KNeighborsClassifier(), X_train,
```

```
Y_train, cv=kfold, scoring=scoring)\n",
        "\n",
        "knn = KNeighborsClassifier()\n",
        "knn.fit(X_train, Y_train)\n",
        "predictions = knn.predict(X_validation)\n",
        "print(accuracy_score(Y_validation, predictions))\n",
        "print(confusion_matrix(Y_validation, predictions))\n",
        "print(classification_report(Y_validation, predictions))\n",
        "\n",
        "print(\"%.2f  %.4f \" % (cv_results.mean(),
cv_results.std()))"
      ],
      "execution_count": 73,
      "outputs": [
        {
          "output_type": "stream",
          "text": [
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_split.py:296: FutureWarning: Setting a random_state
has no effect since shuffle is False. This will raise an error in
0.24. You should leave random_state to its default (None), or set
shuffle=True.\n",
            "  FutureWarning\n",
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_validation.py:536: FitFailedWarning: Estimator fit
failed. The score on this train-test partition for these parameters
will be set to nan. Details: \n",
            "ValueError: could not convert string to float:
'SepalLengthCm'\n",
            "\n",
            "  FitFailedWarning)\n",
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_validation.py:536: FitFailedWarning: Estimator fit
failed. The score on this train-test partition for these parameters
will be set to nan. Details: \n",
            "ValueError: could not convert string to float:
'SepalLengthCm'\n",
            "\n",
            "  FitFailedWarning)\n",
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_validation.py:536: FitFailedWarning: Estimator fit
failed. The score on this train-test partition for these parameters
will be set to nan. Details: \n",
            "ValueError: could not convert string to float:
'SepalLengthCm'\n",
            "\n",
            "  FitFailedWarning)\n",
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_validation.py:536: FitFailedWarning: Estimator fit
failed. The score on this train-test partition for these parameters
```

```
will be set to nan. Details: \n",
            "ValueError: could not convert string to float:
'SepalLengthCm'\n",
            "\n",
            "  FitFailedWarning)\n",
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_validation.py:536: FitFailedWarning: Estimator fit
failed. The score on this train-test partition for these parameters
will be set to nan. Details: \n",
            "ValueError: could not convert string to float:
'SepalLengthCm'\n",
            "\n",
            "  FitFailedWarning)\n",
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_validation.py:536: FitFailedWarning: Estimator fit
failed. The score on this train-test partition for these parameters
will be set to nan. Details: \n",
            "ValueError: could not convert string to float:
'SepalLengthCm'\n",
            "\n",
            "  FitFailedWarning)\n",
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_validation.py:536: FitFailedWarning: Estimator fit
failed. The score on this train-test partition for these parameters
will be set to nan. Details: \n",
            "ValueError: could not convert string to float:
'SepalLengthCm'\n",
            "\n",
            "  FitFailedWarning)\n",
            "/usr/local/lib/python3.7/dist-packages/sklearn/
model_selection/_validation.py:536: FitFailedWarning: Estimator fit
failed. The score on this train-test partition for these parameters
will be set to nan. Details: \n",
            "ValueError: could not convert string to float:
'SepalLengthCm'\n",
            "\n",
            "  FitFailedWarning)\n"
          ],
          "name": "stderr"
        },
        {
          "output_type": "error",
          "ename": "ValueError",
          "evalue": "ignored",
          "traceback": [

"\u001b[0;31m---------------------------------------------------------
------------------\u001b[0m",
            "\u001b[0;31mValueError\u001b[0m
Traceback (most recent call last)",
```

```
            "\u001b[0;32m<ipython-input-73-0652b290d415>\u001b[0m in
\u001b[0;36m<module>\u001b[0;34m()\u001b[0m\n\u001b[1;32m
1\u001b[0m \u001b[0mkfold\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mmodel_selection\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0mKFold\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mn_split
s\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;36m10\u001b[0m\u001b[0;34m,
\u001b[0m
\u001b[0mrandom_state\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mseed\u001
b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
;32m----> 2\u001b[0;31m \u001b[0mcv_results\u001b[0m
\u001b[0;34m=\u001b[0m \u001b[0mmodel_selection\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0mcross_val_score\u001b[0m\u001b[0;34m(\u001b[0m\u001b
[0mKNeighborsClassifier\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mX_train\u001b[0m\u001b[0;34m,
\u001b[0m \u001b[0mY_train\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mcv\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mkfold\u001b[0m\u001b
b[0;34m,\u001b[0m
\u001b[0mscoring\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mscoring\u001b[
0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m      3\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m      4\u001b[0m
\u001b[0mknn\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mKNeighborsClassifier\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;3
4m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m      5\u001b[0m \u001b[0mknn\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0mfit\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mX_train\u
001b[0m\u001b[0;34m,\u001b[0m \u001b[0mY_train\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
            "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/model_selection/_validation.py\u001b[0m in
\u001b[0;36mcross_val_score\u001b[0;34m(estimator, X, y, groups,
scoring, cv, n_jobs, verbose, fit_params, pre_dispatch, error_score)
\u001b[0m\n\u001b[1;32m    388\u001b[0m
\u001b[0mfit_params\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mfit_params\
u001b[0m\u001b[0;34m,
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    389\u001b[0m
\u001b[0mpre_dispatch\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mpre_dispa
tch\u001b[0m\u001b[0;34m,
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
;32m--> 390\u001b[0;31m
error_score=error_score)\n\u001b[0m\u001b[1;32m    391\u001b[0m
\u001b[0;32mreturn\u001b[0m
\u001b[0mcv_results\u001b[0m\u001b[0;34m[\u001b[0m\u001b[0;34m'test_sc
ore'\u001b[0m\u001b[0;34m]
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    392\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n",
```

```
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/model_selection/_validation.py\u001b[0m in
\u001b[0;36mcross_validate\u001b[0;34m(estimator, X, y, groups,
scoring, cv, n_jobs, verbose, fit_params, pre_dispatch,
return_train_score, return_estimator, error_score)
\u001b[0m\n\u001b[1;32m    234\u001b[0m
\u001b[0mreturn_times\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;32mTrue\u
001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mreturn_estimator\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mretur
n_estimator\u001b[0m\u001b[0;34m,
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    235\u001b[0m             error_score=error_score)
\n\u001b[0;32m--> 236\u001b[0;31m         for train, test in
cv.split(X, y, groups))\n\u001b[0m\u001b[1;32m    237\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m    238\u001b[0m
\u001b[0mzipped_scores\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mlist\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mzip\u001b[0m\u001
b[0;34m(\u001b[0m\u001b[0;34m*\u001b[0m\u001b[0mscores\u001b[0m\u001b[
0;34m)\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
joblib/parallel.py\u001b[0m in \u001b[0;36m__call__\u001b[0;34m(self,
iterable)\u001b[0m\n\u001b[1;32m   1042\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0m_iterating\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0m_original_iterator\u001b[0m \u001b[0;32mis\u001b[0m
\u001b[0;32mnot\u001b[0m
\u001b[0;32mNone\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u0
01b[0m\n\u001b[1;32m   1043\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m-> 1044\u001b[0;31m
\u001b[0;32mwhile\u001b[0m \u001b[0mself\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0mdispatch_one_batch\u001b[0m\u001b[0;34m(\u001b[0m\u0
01b[0miterator\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m   1045\u001b[0m
\u001b[0;32mpass\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u0
01b[0m\n\u001b[1;32m   1046\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
joblib/parallel.py\u001b[0m in
\u001b[0;36mdispatch_one_batch\u001b[0;34m(self, iterator)
\u001b[0m\n\u001b[1;32m    857\u001b[0m
\u001b[0;32mreturn\u001b[0m
\u001b[0;32mFalse\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u
001b[0m\n\u001b[1;32m    858\u001b[0m
\u001b[0;32melse\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
;32m--> 859\u001b[0;31m
\u001b[0mself\u001b[0m\u001b[0;34m.
```

\u001b[0m\u001b[0m_dispatch\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mtas
ks\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m   860\u001b[0m
\u001b[0;32mreturn\u001b[0m
\u001b[0;32mTrue\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u0
01b[0m\n\u001b[1;32m   861\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n",
            "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
joblib/parallel.py\u001b[0m in \u001b[0;36m_dispatch\u001b[0;34m(self,
batch)\u001b[0m\n\u001b[1;32m   775\u001b[0m
\u001b[0;32mwith\u001b[0m \u001b[0mself\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0m_lock\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m   776\u001b[0m   \u001b[0mjob_idx\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0mlen\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\u001b[0m\u001
b[0;34m.\u001b[0m\u001b[0m_jobs\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
;32m--> 777\u001b[0;31m   \u001b[0mjob\u001b[0m
\u001b[0;34m=\u001b[0m \u001b[0mself\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0m_backend\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0mapply_async\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mb
atch\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mcallback\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mcb\u001b[0m\u
001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m   778\u001b[0m   \u001b[0;31m# A job can
complete so quickly than its callback
is\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001
b[0m\u001b[0m\n\u001b[1;32m   779\u001b[0m   \u001b[0;31m#
called before we get here, causing self._jobs
to\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001
b[0m\u001b[0m\n",
            "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
joblib/_parallel_backends.py\u001b[0m in
\u001b[0;36mapply_async\u001b[0;34m(self, func, callback)
\u001b[0m\n\u001b[1;32m   206\u001b[0m   \u001b[0;32mdef\u001b[0m
\u001b[0mapply_async\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\u001b
[0m\u001b[0;34m,\u001b[0m \u001b[0mfunc\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mcallback\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;32mNone\u001b
[0m\u001b[0;34m)\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m   207\u001b[0m   \u001b[0;34m\"\"\"Schedule a func to be
run\"\"\"\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\
n\u001b[0;32m--> 208\u001b[0;31m   \u001b[0mresult\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0mImmediateResult\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mfunc\u
001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0

```
m\u001b[1;32m    209\u001b[0m         \u001b[0;32mif\u001b[0m
\u001b[0mcallback\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    210\u001b[0m
\u001b[0mcallback\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mresult\u001b[
0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
joblib/_parallel_backends.py\u001b[0m in
\u001b[0;36m__init__\u001b[0;34m(self, batch)\u001b[0m\n\u001b[1;32m
570\u001b[0m         \u001b[0;31m# Don't delay the application, to
avoid keeping the
input\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u
001b[0m\u001b[0m\n\u001b[1;32m    571\u001b[0m         \u001b[0;31m#
arguments in
memory\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\
u001b[0m\u001b[0m\n\u001b[0;32m--> 572\u001b[0;31m
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mresults\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0mbatch\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m    573\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m    574\u001b[0m
\u001b[0;32mdef\u001b[0m
\u001b[0mget\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\u001b[0m\u001
b[0;34m)\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
joblib/parallel.py\u001b[0m in \u001b[0;36m__call__\u001b[0;34m(self)
\u001b[0m\n\u001b[1;32m    261\u001b[0m
\u001b[0;32mwith\u001b[0m
\u001b[0mparallel_backend\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\
u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m_backend\u001b[0m\u001b[0;34m,
\u001b[0m
\u001b[0mn_jobs\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mself\u001b[0m\u
001b[0;34m.\u001b[0m\u001b[0m_n_jobs\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    262\u001b[0m             return [func(*args, **kwargs)
\n\u001b[0;32m--> 263\u001b[0;31m                     for func, args,
kwargs in self.items]\n\u001b[0m\u001b[1;32m    264\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m    265\u001b[0m
\u001b[0;32mdef\u001b[0m
\u001b[0m__reduce__\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\u001b[
0m\u001b[0;34m)\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
joblib/parallel.py\u001b[0m in
\u001b[0;36m<listcomp>\u001b[0;34m(.0)\u001b[0m\n\u001b[1;32m
261\u001b[0m         \u001b[0;32mwith\u001b[0m
```

\u001b[0mparallel_backend\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\
u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m_backend\u001b[0m\u001b[0;34m,
\u001b[0m
\u001b[0mn_jobs\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mself\u001b[0m\u
001b[0;34m.\u001b[0m\u001b[0m_n_jobs\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m   262\u001b[0m   return [func(*args, **kwargs)
\n\u001b[0;32m--> 263\u001b[0;31m   for func, args,
kwargs in self.items]\n\u001b[0m\u001b[1;32m   264\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m   265\u001b[0m
\u001b[0;32mdef\u001b[0m
\u001b[0m__reduce__\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\u001b[
0m\u001b[0;34m)\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/model_selection/_validation.py\u001b[0m in
\u001b[0;36m_fit_and_score\u001b[0;34m(estimator, X, y, scorer, train,
test, verbose, parameters, fit_params, return_train_score,
return_parameters, return_n_test_samples, return_times,
return_estimator, error_score)\u001b[0m\n\u001b[1;32m   542\u001b[0m
\u001b[0;32melse\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m   543\u001b[0m   \u001b[0mfit_time\u001b[0m
\u001b[0;34m=\u001b[0m \u001b[0mtime\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0mtime\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)
\u001b[0m \u001b[0;34m-\u001b[0m
\u001b[0mstart_time\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m
\u001b[0m\n\u001b[0;32m--> 544\u001b[0;31m
\u001b[0mtest_scores\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0m_score\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mestimator\u001b
[0m\u001b[0;34m,\u001b[0m \u001b[0mX_test\u001b[0m\u001b[0;34m,
\u001b[0m \u001b[0my_test\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mscorer\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m   545\u001b[0m   \u001b[0mscore_time\u001b[0m
\u001b[0;34m=\u001b[0m \u001b[0mtime\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0mtime\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)
\u001b[0m \u001b[0;34m-\u001b[0m \u001b[0mstart_time\u001b[0m
\u001b[0;34m-\u001b[0m
\u001b[0mfit_time\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u
001b[0m\n\u001b[1;32m   546\u001b[0m   \u001b[0;32mif\u001b[0m
\u001b[0mreturn_train_score\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/model_selection/_validation.py\u001b[0m in
\u001b[0;36m_score\u001b[0;34m(estimator, X_test, y_test, scorer)
\u001b[0m\n\u001b[1;32m   589\u001b[0m
\u001b[0mscores\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mscorer\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mestimator\u001b

```
[0m\u001b[0;34m,\u001b[0m \u001b[0mX_test\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    590\u001b[0m     \u001b[0;32melse\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
;32m--> 591\u001b[0;31m         \u001b[0mscores\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0mscorer\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mestimator\u001b
[0m\u001b[0;34m,\u001b[0m \u001b[0mX_test\u001b[0m\u001b[0;34m,
\u001b[0m \u001b[0my_test\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m    592\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m    593\u001b[0m
error_msg = (\"scoring must return a number, got %s (%s) \"\n",
            "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/metrics/_scorer.py\u001b[0m in
\u001b[0;36m__call__\u001b[0;34m(self, estimator, *args, **kwargs)
\u001b[0m\n\u001b[1;32m     85\u001b[0m
\u001b[0;32mif\u001b[0m
\u001b[0misinstance\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mscorer\u001
b[0m\u001b[0;34m,\u001b[0m \u001b[0m_BaseScorer\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m     86\u001b[0m                 score =
scorer._score(cached_call, estimator,\n\u001b[0;32m---> 87\u001b[0;31m
*args, **kwargs)\n\u001b[0m\u001b[1;32m     88\u001b[0m
\u001b[0;32melse\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m     89\u001b[0m                 \u001b[0mscore\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0mscorer\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mestimator\u001b
[0m\u001b[0;34m,\u001b[0m
\u001b[0;34m*\u001b[0m\u001b[0margs\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0;34m**\u001b[0m\u001b[0mkwargs\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
            "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/metrics/_scorer.py\u001b[0m in
\u001b[0;36m_score\u001b[0;34m(self, method_caller, estimator, X,
y_true, sample_weight)\u001b[0m\n\u001b[1;32m    203\u001b[0m
\"\"\"\n\u001b[1;32m    204\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m--> 205\u001b[0;31m
\u001b[0my_pred\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mmethod_caller\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mestimato
r\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0;34m\"predict\"\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mX\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m    206\u001b[0m
\u001b[0;32mif\u001b[0m
\u001b[0msample_weight\u001b[0m \u001b[0;32mis\u001b[0m
\u001b[0;32mnot\u001b[0m \u001b[0;32mNone\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
```

```
;32m    207\u001b[0m             return self._sign *
self._score_func(y_true, y_pred,\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/metrics/_scorer.py\u001b[0m in
\u001b[0;36m_cached_call\u001b[0;34m(cache, estimator, method, *args,
**kwargs)\u001b[0m\n\u001b[1;32m     50\u001b[0m
\u001b[0;34m\"\"\"Call estimator with method and args and kwargs.
\"\"\"\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u
001b[1;32m     51\u001b[0m     \u001b[0;32mif\u001b[0m
\u001b[0mcache\u001b[0m \u001b[0;32mis\u001b[0m
\u001b[0;32mNone\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
;32m---> 52\u001b[0;31m         \u001b[0;32mreturn\u001b[0m
\u001b[0mgetattr\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mestimator\u001
b[0m\u001b[0;34m,\u001b[0m \u001b[0mmethod\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m*\u001b[0m\u001b[0margs\u00
1b[0m\u001b[0;34m,\u001b[0m
\u001b[0;34m**\u001b[0m\u001b[0mkwargs\u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m     53\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m     54\u001b[0m
\u001b[0;32mtry\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/neighbors/_classification.py\u001b[0m in
\u001b[0;36mpredict\u001b[0;34m(self, X)\u001b[0m\n\u001b[1;32m
169\u001b[0m             \u001b[0mClass\u001b[0m
\u001b[0mlabels\u001b[0m \u001b[0;32mfor\u001b[0m
\u001b[0meach\u001b[0m \u001b[0mdata\u001b[0m
\u001b[0msample\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    170\u001b[0m         \"\"\"\n\u001b[0;32m--> 171\u001b[0;31m
\u001b[0mX\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mcheck_array\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mX\u001b[0m
\u001b[0;34m,\u001b[0m
\u001b[0maccept_sparse\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;34m'csr'
\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m    172\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m    173\u001b[0m
\u001b[0mneigh_dist\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mneigh_ind\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mself\u001b[0;34m.
\u001b[0m\u001b[0mkneighbors\u001b[0m\u001b[0;34m(\u001b[0mX\
u001b[0m\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/
sklearn/utils/validation.py\u001b[0m in
\u001b[0;36mcheck_array\u001b[0;34m(array, accept_sparse,
accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d,
```

```
allow_nd, ensure_min_samples, ensure_min_features, warn_on_dtype,
estimator)\u001b[0m\n\u001b[1;32m    529\u001b[0m
\u001b[0marray\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0marray\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0mastype\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mdtype\
u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mcasting\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;34m\"unsafe\"\
u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mcopy\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;32mFalse\u001b[0m
\u001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    530\u001b[0m
\u001b[0;32melse\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
;32m--> 531\u001b[0;31m                    \u001b[0marray\u001b[0m
\u001b[0;34m=\u001b[0m \u001b[0mnp\u001b[0m\u001b[0;34m.
\u001b[0m\u001b[0masarray\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0marray
\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0morder\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0morder\u001b[0m\u
001b[0;34m,\u001b[0m
\u001b[0mdtype\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mdtype\u001b[0m\u
001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m    532\u001b[0m             \u001b[0;32mexcept\u001b[0m
\u001b[0mComplexWarning\u001b[0m\u001b[0;34m:
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1
;32m    533\u001b[0m                 raise ValueError(\"Complex data
not supported\\n\"\n",
           "\u001b[0;32m/usr/local/lib/python3.7/dist-packages/numpy/
core/_asarray.py\u001b[0m in \u001b[0;36masarray\u001b[0;34m(a, dtype,
order)\u001b[0m\n\u001b[1;32m     81\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m     82\u001b[0m
\"\"\"\n\u001b[0;32m---> 83\u001b[0;31m
\u001b[0;32mreturn\u001b[0m
\u001b[0marray\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0ma\u001b[0m\u001b
[0;34m,\u001b[0m \u001b[0mdtype\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mcopy\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;32mFalse\u001b[0m
\u001b[0;34m,\u001b[0m
\u001b[0morder\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0morder\u001b[0m\u
001b[0;34m)
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m     84\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m     85\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n",
           "\u001b[0;31mValueError\u001b[0m: could not convert string
to float: 'SepalLengthCm'"
          ]
        }
      ]
    }
```

```
    ]
}
```