**IDS PROJECT REPORT**

**IRIS ML CLASSIFICATION**

SUBMITTED BY GROUP 27

RISHABH SAHU 19UCS129

MANIT MEHTA 19UCS147

ADITYA KHANDELWAL             19UCS163

ABHISHEK SINGHAL 19UCS127

Overview

In this project we are applying ML classification on IRIS dataset that is imported from

<https://archive.ics.uci.edu/ml/datasets/Iris> .Iris is a genus of 260–300 species of flowering plants with showy flowers. It takes its name from the Greek word for a rainbow, which is also the name for the Greek goddess of the rainbow, Iris. The flowers commonly possess three sepals, three petals, and three broad pollen-receptive stigma branches, under which the pollen-producing anthers are hidden.  The dataset contains Sepal lengths , Sepal widths ,Petal lengths and petal widths as features of the Iris plants along with their class names corresponding to the species. Iris plant has three main species which are Iris setosa, Iris versicolor, and Iris virginica. This project aims to create a Machine learning classification model that can classify a data point containing sepal and petal characteristics to deduce the species of that IRIS flower from among these 3 species.

Our questions regarding the data

Do the different species of Iris plants show a clear demarcation in their sepal and petal characteristics?

Can we perform ML classification on the data set with high accuracy?

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Iris setosa | Iris versicolor | Iris virginica |

Code for importing the data  
Importing the data

columns = ['Sepal length', 'Sepal width', 'Petal length', 'Petal width', 'Class\_labels']

#get data from url

frameData=pd.read\_csv('<https://archive.ics.uci.edu/ml/machine-learning-databases/iris/bezdekIris.data',names=columns>)

#data values collected in numpy array

dataArray=frameData.value

Preliminary analysis

# Theory

First we find out the important statistics about the data using the measures of central tendencies which are the mean ,median and mode.We performed this task by using the functions of pandas python library which are applied on the pandas dataframe object.

# Code for preliminary analysis

Display data in tabular form

df = pd.DataFrame(frameData)

print("\n-----------  Mean of complete dataset -----------\n")

display(df.mean())

print("\n----------- Median of complete dataset -----------\n")

display(df.median())

print("\n----------- Mode of complete dataset  -----------\n")

display(df.mode())

grouped = df.groupby(df.Class\_labels)

Isetosa = grouped.get\_group("Iris-setosa")

Iversicolor = grouped.get\_group("Iris-versicolor")

Ivirginica = grouped.get\_group("Iris-virginica")

print("\n-----------  Dataset -----------\n")

display(Isetosa.head())

display(Iversicolor.head())

display(Ivirginica.head())

print("-----------Mean,Median & Mode of Iris-setosa -----------\n")

print("-----Mean-----")

display(Isetosa.mean())

print("-----Median-----")

display(Isetosa.median())

print("-----Mode-----")

display(Isetosa.mode())

print("-----------Mean,Median & Mode of Iris-Iris-versicolor -----------\n")

print("-----Mean-----")

display(Iversicolor.mean())

print("-----Median-----")

display(Iversicolor.median())

print("-----Mode-----")

display(Iversicolor.mode())

print("-----------Mean,Median & Mode of Iris-virginica -----------\n")

print("-----Mean-----")

display(Ivirginica.mean())

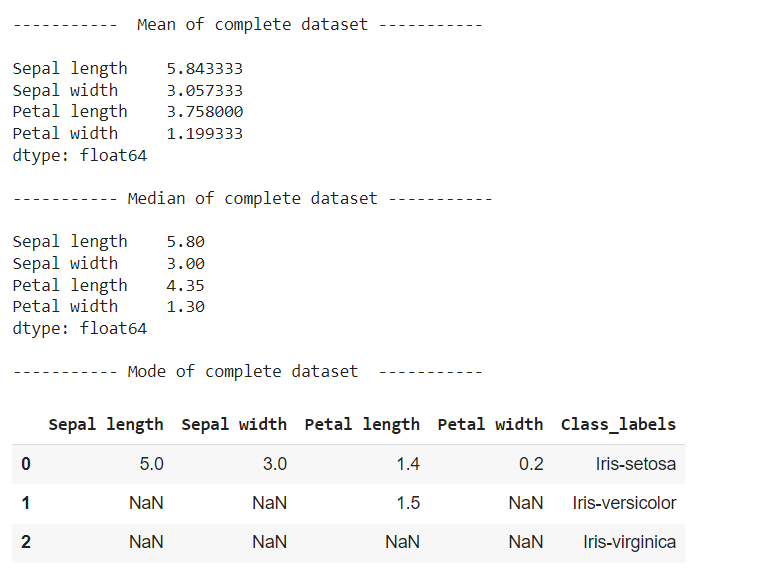
print("-----Median-----")

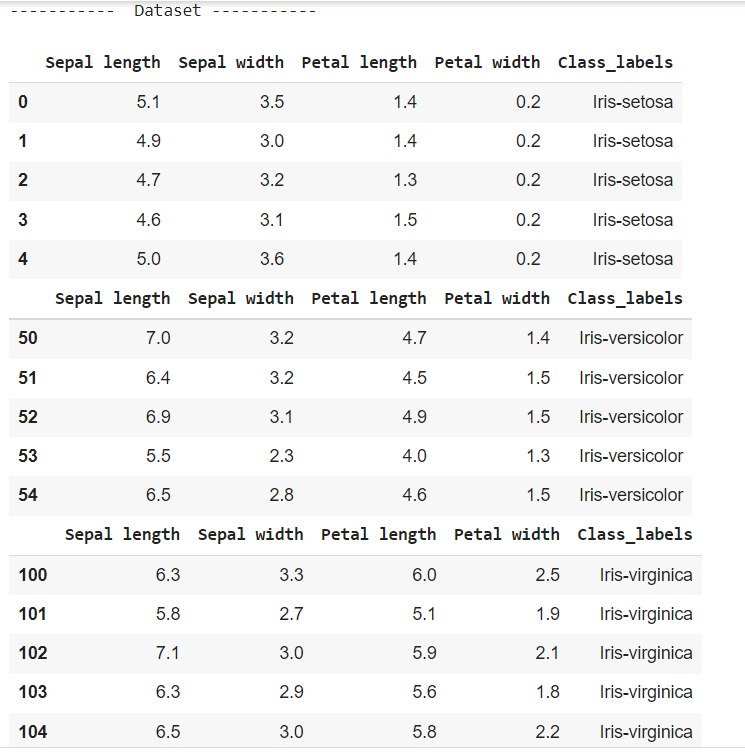
display(Ivirginica.median())

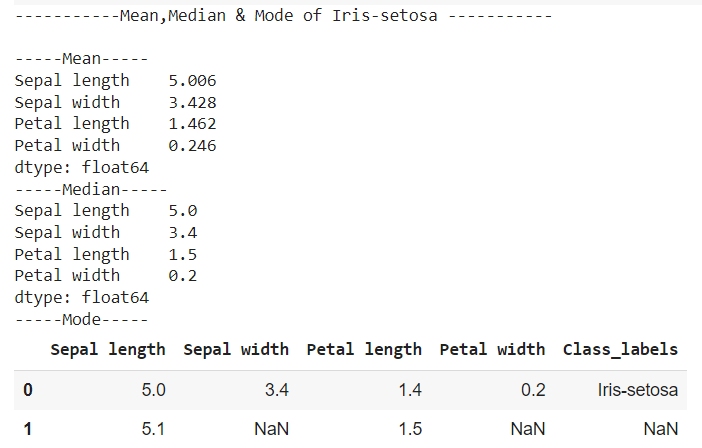
print("-----Mode-----")

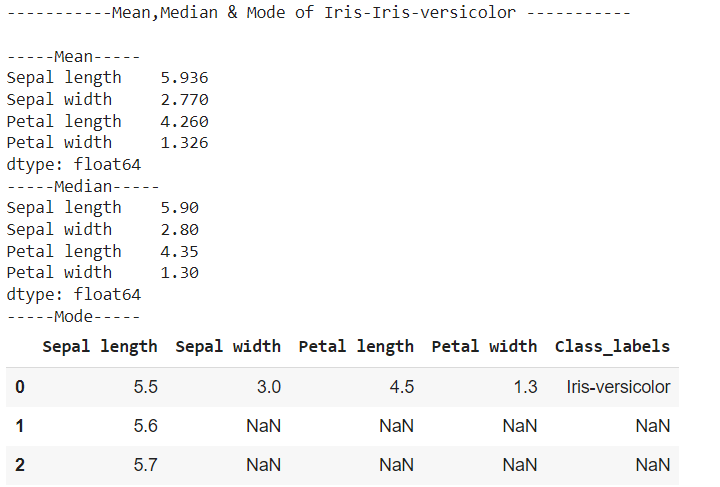
display(Ivirginica.mode())

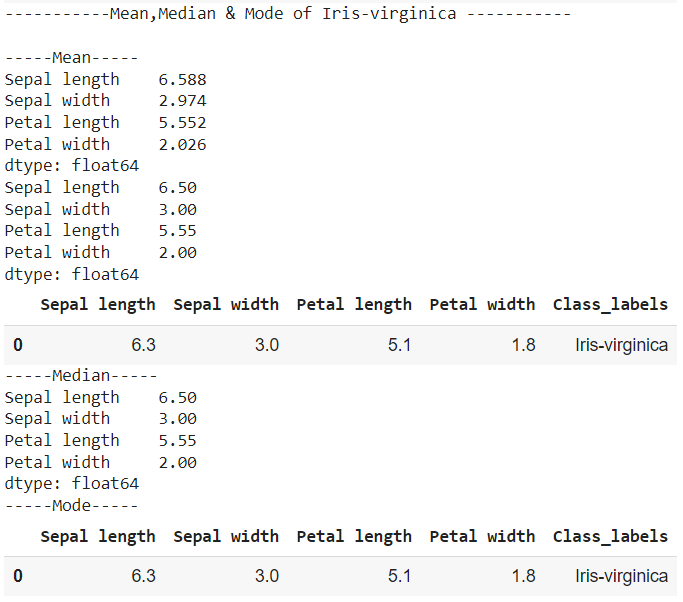
Results of preliminary analysis











# Inferences

          On analysing the data we found –

1) The sepal length mean value is maximum for iris verginica . However, iris verginica and iris  versicolor both have sepal length mean value greater than the overall mean of the complete dataset .

2) iris setosa has the sepal width mean value greater than the overall mean of the complete dataset .

3) Iris verginica has the maximum petal length mean value .However iris verginica and iris versicolor both have petal  length mean value greater than the overall mean of the complete dataset .

4) Iris verginica has the maximum petal width mean value. However, iris  verginica and iris  versicolor both have petal  width mean value greater than the overall mean of the complete dataset.

Graphical Analysis

# Code and procedure for graphical analysis

We used the matplotlib library to display the data points in scatter plot format taking 2 features at a time.

import matplotlib.patches as mpatches

print("\n Iris-setosa-Red   Iris-versico;our- Green   Iris-virginica-Yellow\n")

# create a figure and axis

colors = {'Iris-setosa':'red', 'Iris-versicolor':'green', 'Iris-virginica':'yellow'}

fig, ax = plt.subplots()

# scatter the sepal\_length against the sepal\_width

for i in range(len(frameData['Sepal length'])):

  scatter=ax.scatter(frameData['Sepal length'][i], frameData['Sepal width'][i],color=colors[frameData['Class\_labels'][i]],label=frameData['Class\_labels'][i])

#display legend

red\_patch = mpatches.Patch(color='red', label='Iris-setosa')

yellow\_patch = mpatches.Patch(color='yellow', label='Iris-virginica')

green\_patch = mpatches.Patch(color='green', label='Iris-versicolor')

plt.legend(handles=[red\_patch,green\_patch,yellow\_patch],loc="lower right")

# set a title and labels

ax.set\_title('Iris Dataset')

ax.set\_xlabel('Sepal length')

ax.set\_ylabel('Sepal width')

# create a figure and axis

colors = {'Iris-setosa':'red', 'Iris-versicolor':'green', 'Iris-virginica':'yellow'}

fig, ax = plt.subplots()

# scatter the sepal\_length against the sepal\_width

for i in range(len(frameData['Petal length'])):

  ax.scatter(frameData['Petal length'][i], frameData['Petal width'][i],color=colors[frameData['Class\_labels'][i]])

# set a title and labels

ax.set\_title('Iris Dataset')

ax.set\_xlabel('Petal length')

ax.set\_ylabel('Petal width')

#display legend

red\_patch = mpatches.Patch(color='red', label='Iris-setosa')

yellow\_patch = mpatches.Patch(color='yellow', label='Iris-virginica')

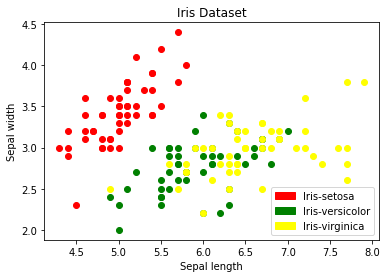
green\_patch = mpatches.Patch(color='green', label='Iris-versicolor')

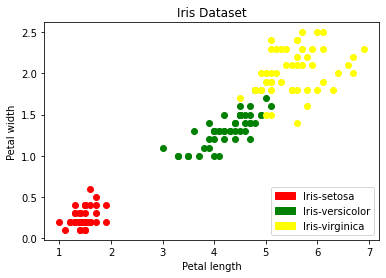
plt.legend(handles=[red\_patch,green\_patch,yellow\_patch],loc="lower right")

# Results

Iris-setosa-Red Iris-versico;our- Green Iris-virginica-Yellow

<matplotlib.legend.Legend at 0x7f9dab720b10>





# Inferences from plots

While the in sepal length and sepal width are not making clear separations among the classes but there are some inferences in petal length and petal width which are separating the classes clearly :

(a).In Iris setosa petal length mostly lies between [1,2] and petal width between [0,1]

(b).In Iris versicolor petal length mostly lies between [3,5] and petal width between[1,2]

            (c).In iris virginica petal length mostly lies between [5,7] and petal width between[1.5,2.5]

ML classification

# Theory

We have used Support Vector Machine algorithm to classify our data. It is a discriminant based classification method. In the classification of two classes, support vectors are the cases that are close to the boundary . These are the uncertain or erroneous cases that lie in the vicinity of the boundary between two classes. The optimal separating hyperplane for the two classes is deduced from the support vectors by training its parameters ,i.e. the weights to be such that the margin of separation between any two classes is minimised.It is based on the Lagrange multipliers based constrained optimisation approach. For classifying more than two classes multiple such classifiers are found corresponding to each pair of classes.

# Reasons and justifications for choosing Support vector machine classification

From the plots derived earlier it is clear that the data points of each class are clustered together and we can have a clear boundary of separation between classes. Thus applying SVM is possible ,and we can apply the algorithm once for creating the prediction model instead of applying it again and again for classifying each data point (which would have happened if we tried to use k nearest neighbours based classification) .

Code and procedure

We have used scikit-learn library for applying SVM.We also used the numpy library to store the data values as well as pass them as parameters to scikit -learn library functions. The related code snippets are shown below.

#importing support vector classifier from scikit-learn library

from sklearn.svm import SVC

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

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import ConfusionMatrixDisplay

from sklearn.metrics import classification\_report

#separating the data feature categories from the feature values

features = dataArray[:,0:4]#features

categories = dataArray[:,4]#corresponding class names

#separate a subset of provided data into training and test data for classifier evaluation

feature\_train, feature\_test, category\_train,category\_test = train\_test\_split(features, categories, test\_size=0.5)

# Create classification model

svn = SVC()

svn.fit(feature\_train, category\_train)

#get predictions on test data from original dataset

predictions = svn.predict(feature\_test)

# Calculate the accuracy with respect to test subset of the data

accuracy\_score(category\_test, predictions)

#display confusion matrix from classification on the provided dataset values

cm=confusion\_matrix(category\_test,predictions)

cm\_display = ConfusionMatrixDisplay(cm).plot()

# A detailed classification report

print(classification\_report(category\_test, predictions))

#Applying classification on some new patterns data values

#Sepal length ,Sepal width, Petal length ,Petal Width

feature\_new = np.array([[3, 2, 1, 0.2], [  4.9, 2.2, 3.8, 1.1 ], [  5.3, 2.5, 4.6, 1.9 ]])

#Prediction of the species from the input vector

prediction = svn.predict(feature\_new)

print("Prediction of Species by Support vector machine for data points : {}".format(prediction))

Results

|  |
| --- |
| precision    recall  f1-score   support      Iris-setosa       1.00      1.00      1.00        23  Iris-versicolor       0.89      1.00      0.94        24   Iris-virginica       1.00      0.89      0.94        28         accuracy                           0.96        75        macro avg       0.96      0.96      0.96        75     weighted avg       0.96      0.96      0.96        75        Prediction of Species by Support vector machine for data points : ['Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'] |

Inferences from ML classification

Our classifier is able to classify data points with high accuracy on the provided dataset.This also tells us that only very few of the iris flowers from the sample data have their petal and sepal characteristics quite different from their class ,which is about 3 of the samples and rest all other flowers have their petals and sepal characteristics following a natural similarity with other flowers in the same class.

Conclusive remarks

We were able to apply ML classification on the Iris dataset with high accuracy. The procedures and results also answered our questions related to the dataset. As we were able to classify the Iris plants with high accuracy ,and the Petal lengths and widths show clear demarcation among the three classes. Sepal lengths and widths do show a satisfactory demarcation . The use of Support Vector Machine classifier proved to be beneficial as well as time saving as we didn’t have to apply the algorithm again and again on each data point.

References

<https://scikit-learn.org/>

<https://numpy.org/doc/stable/reference/>

[Users guide — Matplotlib 3.5.1 documentation](https://matplotlib.org/stable/users/index.html)

<https://pandas.pydata.org/docs/>

[Iris (plant) - Wikipedia](https://en.wikipedia.org/wiki/Iris_(plant))