9

**NAME – Sumit Dadarao Ingale**

**DATA SCIENCE INTERNSHIP @ OASIS INFOBYTE**

**TASK 1**

**PROJECT NAME - IRIS FLOWER CLASSIFICATION**

In [ ]:

from IPython.display import Image

Image(url='https://editor.analyticsvidhya.com/uploads/51518iris%20img1.png', width=850)

**Problem Statement**

Iris flower has 3 species: Setosa, Versicolor and Virginica, which differs according to their measurements. Now assume that you have the measurements of the iris flowers according to their species, and here your task is to train a machine learning model that can learn from the measurements of the iris species and classify them.

INPUT:  
Sepal Length  
Sepal width  
Petal length  
Petal width \

OUTPUT:  
Class of Flower

In [ ]:

**Importing Libraries**

In [ ]:

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

**Loading the Dataset**

In [ ]:

iris = pd.read\_csv('/content/Iris.csv')

iris.head()

In [ ]:

*# Rename the complex columns name*

iris= iris.rename(columns={'SepalLengthCm':'Sepal\_Length',

'SepalWidthCm':'Sepal\_Width',

'PetalLengthCm':'Petal\_Length',

'PetalWidthCm':'Petal\_Width'})

In [ ]:

iris.head()

In [ ]:

*# checking null values*

iris.isnull().sum()

In [ ]:

*# checking if the data is biased or not*

iris ['Species'].value\_counts()

In [ ]:

*# checking statistical features*

iris.describe()

**Visualization**

**Scatterplot**

In [ ]:

sns.FacetGrid(iris, hue="Species",height=6).map(plt.scatter,"Petal\_Length","Sepal\_Width").add\_legend()

**Pairplot**

In [ ]:

*# visualize the whole dataset*

sns.pairplot(iris[['Sepal\_Length','Sepal\_Width','Petal\_Length','Petal\_Width','Species']], hue="Species",diag\_kind='kde')

**SEPARATING INPUT COLUMNS AND THE OUTPUT COLUMNS**

In [ ]:

*# Separate features and target*

data=iris.values

*# slicing the matrices*

X=data[:,0:4]

Y=data[:,5]

In [ ]:

print(X.shape)

print(X)

In [ ]:

print(Y.shape)

print(Y)

**SPLITTING DATA INTO TRAINING AND TESTING**

In [ ]:

*# split the data to train and test dataset*

In [ ]:

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

X\_train, X\_test, y\_train, y\_test= train\_test\_split(X,Y, test\_size=0.2)

In [ ]:

print(X\_train.shape)

print(X\_train)

In [ ]:

print(y\_test.shape)

print(y\_test)

In [ ]:

print(X\_test.shape)

print(X\_test)

In [ ]:

print(y\_train.shape)

print(y\_train)

**MODEL 1: SUPPORT VECTOR MACHINE ALGORITHM**

In [ ]:

from sklearn.svm import SVC

model\_svc=SVC()

model\_svc.fit(X\_train,y\_train)

In [ ]:

prediction1 = model\_svc.predict(X\_test)

*#calculate the accuracy*

from sklearn.metrics import accuracy\_score

print(accuracy\_score(y\_test, prediction1))

**MODEL 2: LOGISTIC REGRESSION**

In [ ]:

*# converting categorical variables into numbers*

flower\_mapping = {'Iris-setosa':0,'Iris-versicolor':1,'Iris-virginica':2}

iris['Species']=iris['Species'].map(flower\_mapping)

In [ ]:

iris.head()

In [ ]:

iris.tail()

In [ ]:

*# preparing inputs and outputs*

X=iris [['Sepal\_Length','Sepal\_Width','Petal\_Length','Petal\_Width']].values

y= iris[['Species']].values

In [ ]:

*# LOGISTIC REGRESSION*

from sklearn.linear\_model import LogisticRegression

model= LogisticRegression()

model.fit(X,y)

In [ ]:

*# accuracy*

In [ ]:

model.score(X,y)

In [ ]:

*# make prediction for all 150 species in dataset*

In [ ]:

expected = y

predicted = model.predict(X)

predicted

In [ ]:

*# summarize the fit of the model*

from sklearn import metrics

In [ ]:

print(metrics.classification\_report(expected, predicted))

setosa -> 0  
versicolor -> 1  
virginica ->2

In [ ]:

*# confusion metrics*

print(metrics.confusion\_matrix(expected, predicted))

**MODEL3: DECISION TREE CLASSIFIER**

In [ ]:

from sklearn.tree import DecisionTreeClassifier

model\_DTC = DecisionTreeClassifier()

model\_DTC.fit(X\_train, y\_train)

In [ ]:

prediction3= model\_svc.predict(X\_test)

*#calculate the accuracy*

from sklearn.metrics import accuracy\_score

print(accuracy\_score(y\_test, prediction3))

**New data for prediction**

In [ ]:

*# New data for prediction*

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

*# Predicting the sizes of the iris flowers*

predicted\_sizes = model.predict(X\_new)

*# Output the predicted sizes*

print(predicted\_sizes)

setosa -> 0  
versicolor -> 1  
virginica ->2