devl4a

November 13, 2024

[1]:

**import pandas as pd import numpy as np**

[3]:

df=pd.read\_csv("/content/Iris.csv") df

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [3]: |  | Id | SepalLengthCm | | SepalWidthCm | | PetalLengthCm | | PetalWidthCm | | \ |
|  | 0 | 1 | 5.1 | | 3.5 | | 1.4 | | 0.2 | |  |
|  | 1 | 2 | 4.9 | | 3.0 | | 1.4 | | 0.2 | |  |
|  | 2 | 3 | 4.7 | | 3.2 | | 1.3 | | 0.2 | |  |
|  | 3 | 4 | 4.6 | | 3.1 | | 1.5 | | 0.2 | |  |
|  | 4 | 5 | 5.0 | | 3.6 | | 1.4 | | 0.2 | |  |
| .. | | … … | |  | … |  | … |  | … |  | |
| 145 | | 146 | | 6.7 |  | 3.0 |  | 5.2 |  | 2.3 | |
| 146 | | 147 | | 6.3 |  | 2.5 |  | 5.0 |  | 1.9 | |
| 147 | | 148 | | 6.5 |  | 3.0 |  | 5.2 |  | 2.0 | |
| 148 | | 149 | | 6.2 |  | 3.4 |  | 5.4 |  | 2.3 | |
| 149 | | 150 | | 5.9 |  | 3.0 |  | 5.1 |  | 1.8 | |

[4]:

df.info() df.drop("Id",axis=1,inplace=**True**)

Species

1. Iris-setosa
2. Iris-setosa
3. Iris-setosa
4. Iris-setosa
5. Iris-setosa

.. …

1. Iris-virginica
2. Iris-virginica
3. Iris-virginica
4. Iris-virginica
5. Iris-virginica

[150 rows x 6 columns]

<class 'pandas.core.frame.DataFrame'>

[5]:

RangeIndex: 150 entries, 0 to 149 Data columns (total 6 columns):

# Column Non-Null Count Dtype

1. Id 150 non-null int64
2. SepalLengthCm 150 non-null float64
3. SepalWidthCm 150 non-null float64
4. PetalLengthCm 150 non-null float64
5. PetalWidthCm 150 non-null float64
6. Species 150 non-null object dtypes: float64(4), int64(1), object(1) memory usage: 7.2+ KB

df.describe()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [5]: | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm |
| count | 150.000000 | 150.000000 | 150.000000 | 150.000000 |
| mean | 5.843333 | 3.054000 | 3.758667 | 1.198667 |
| std | 0.828066 | 0.433594 | 1.764420 | 0.763161 |
| min | 4.300000 | 2.000000 | 1.000000 | 0.100000 |
| 25% | 5.100000 | 2.800000 | 1.600000 | 0.300000 |
| 50% | 5.800000 | 3.000000 | 4.350000 | 1.300000 |
| 75% | 6.400000 | 3.300000 | 5.100000 | 1.800000 |
| max | 7.900000 | 4.400000 | 6.900000 | 2.500000 |

[6]:

df.dtypes

* 1. : SepalLengthCm float64 SepalWidthCm float64 PetalLengthCm float64 PetalWidthCm float64

Species object dtype: object

[7]:

df.isnull().sum()

* 1. : SepalLengthCm 0

SepalWidthCm 0

PetalLengthCm 0

PetalWidthCm 0

Species 0

dtype: int64

[8]:

*# Determine X and Y* x=df.iloc[:,0:4].values y=df.iloc[:,4].values

[9]:

*#Label Encoding*

**from sklearn.preprocessing import** LabelEncoder le=LabelEncoder()

y=le.fit\_transform(y)

[11]:

[14]:

*#Training and Testing the model* x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.3,random\_state=0) gaussian=GaussianNB()

gaussian.fit(x\_train,y\_train) y\_pred=gaussian.predict(x\_test) accuracyscore=round(accuracy\_score(y\_test,y\_pred)\*100,2) print(accuracyscore)

precisionscore=precision\_score(y\_test,y\_pred,average='weighted')\*100 print(precisionscore) recallscore=recall\_score(y\_test,y\_pred,average='weighted')\*100 print(recallscore) f1score=f1\_score(y\_test,y\_pred,average='weighted')\*100 print(f1score)

conf\_mat=confusion\_matrix(y\_test,y\_pred) print(conf\_mat) class\_report=classification\_report(y\_test,y\_pred) print(class\_report)

*#Accuracy Prediction*

**from sklearn.metrics import**␣

↪classification\_report,confusion\_matrix,accuracy\_score,precision\_score,recall\_score,f1\_score

*#Model Selection*

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

**from sklearn.naive\_bayes import** GaussianNB

100.0

100.0

100.0

100.0

[[16 0 0]

[ 0 18 0]

[ 0 0 11]]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 1.00 | 1.00 | 1.00 | 16 |
| 1 | 1.00 | 1.00 | 1.00 | 18 |
| 2 | 1.00 | 1.00 | 1.00 | 11 |
| accuracy |  |  | 1.00 | 45 |
| macro avg | 1.00 | 1.00 | 1.00 | 45 |
| weighted avg | 1.00 | 1.00 | 1.00 | 45 |

[ ]: