KNN Program

Date: / /

KNN Program

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,
                            confusion_matrix
from sklearn import datasets

iris = datasets.load_iris()
print(iris)

iris_data = iris.data
iris_labels = iris.target
print(iris_data)
print(iris_labels)
x_train, x_test, y_train, y_test = train_test_split
                    (iris_data, iris_labels, test_size=0.30)

classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)

print('Confusion matrix is as follows')
print(Confusion_matrix(y_test, y_pred))

print("Accuracy metrics")
print(Classification_report(y_test, y_pred))
```

Output justification

$$\begin{bmatrix} 13 & 0 & 0 \\ 0 & 15 & 1 \\ 0 & 2 & 14 \end{bmatrix}$$

Recall = correctly classified / total actual

Label 0 → $\frac{13}{13} = 1$

Label 1 → $\frac{15}{16} = 0.94$

Label 2 → $\frac{14}{16} = 0.88$

Precision = correctly predicted / total predicted

Label 0 → $\frac{13}{13} = 1$

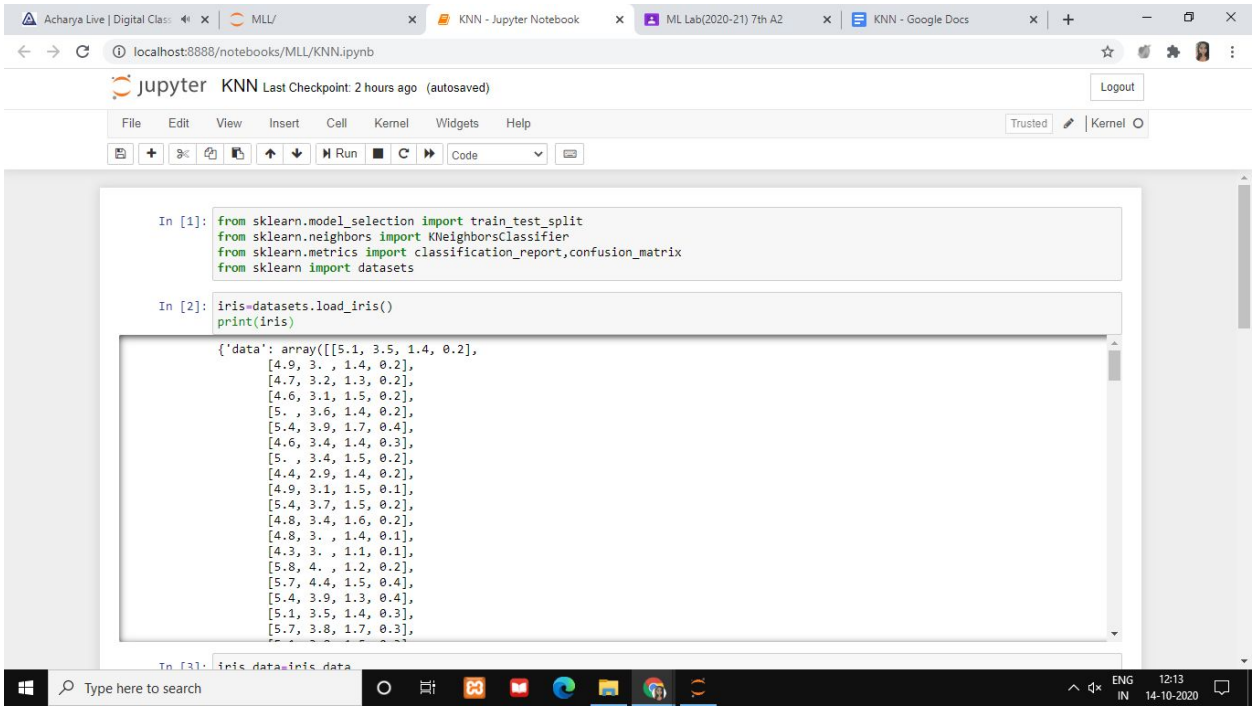
Label 1 → $\frac{15}{17} = 0.88$

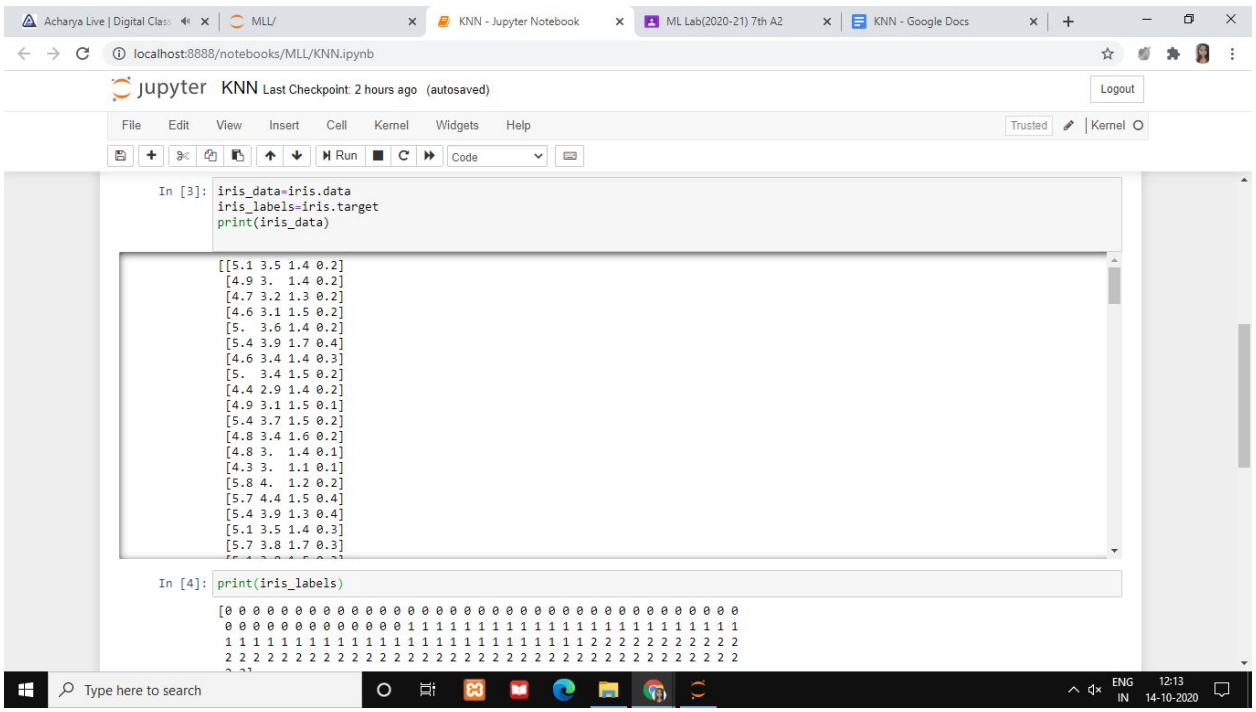
Label 2 → $\frac{14}{15} = 0.93$

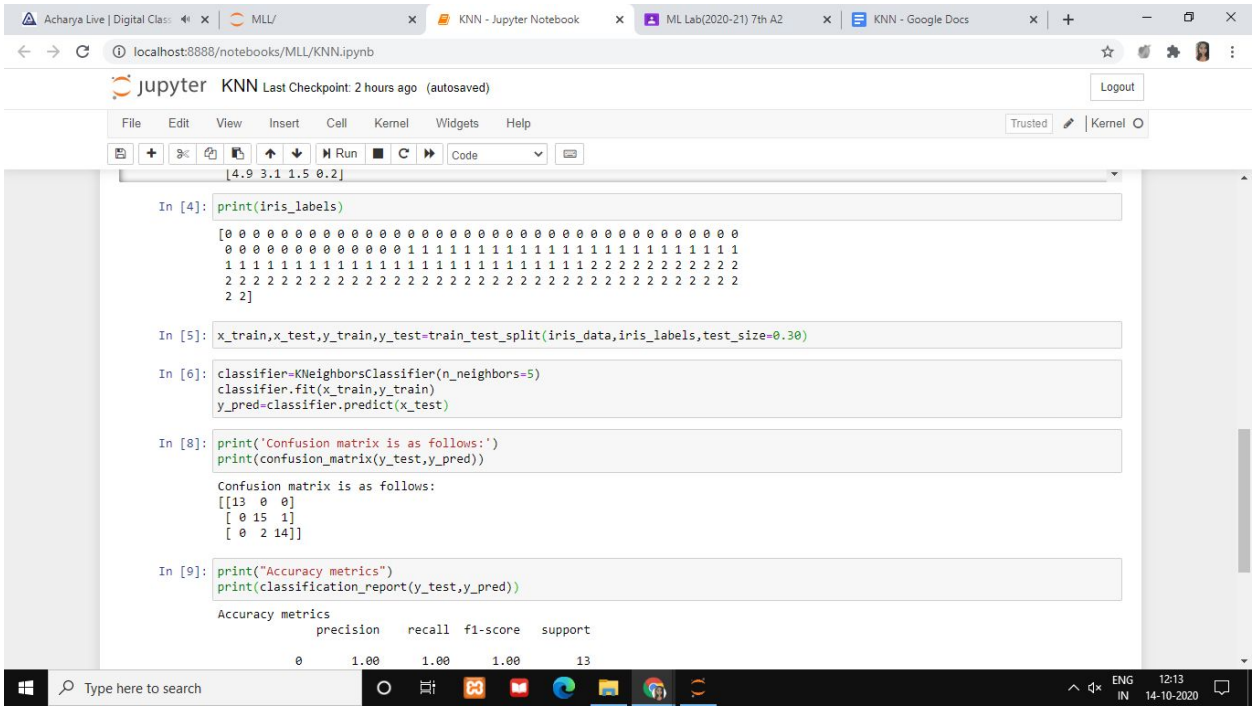
Accuracy = correct prediction / actual value

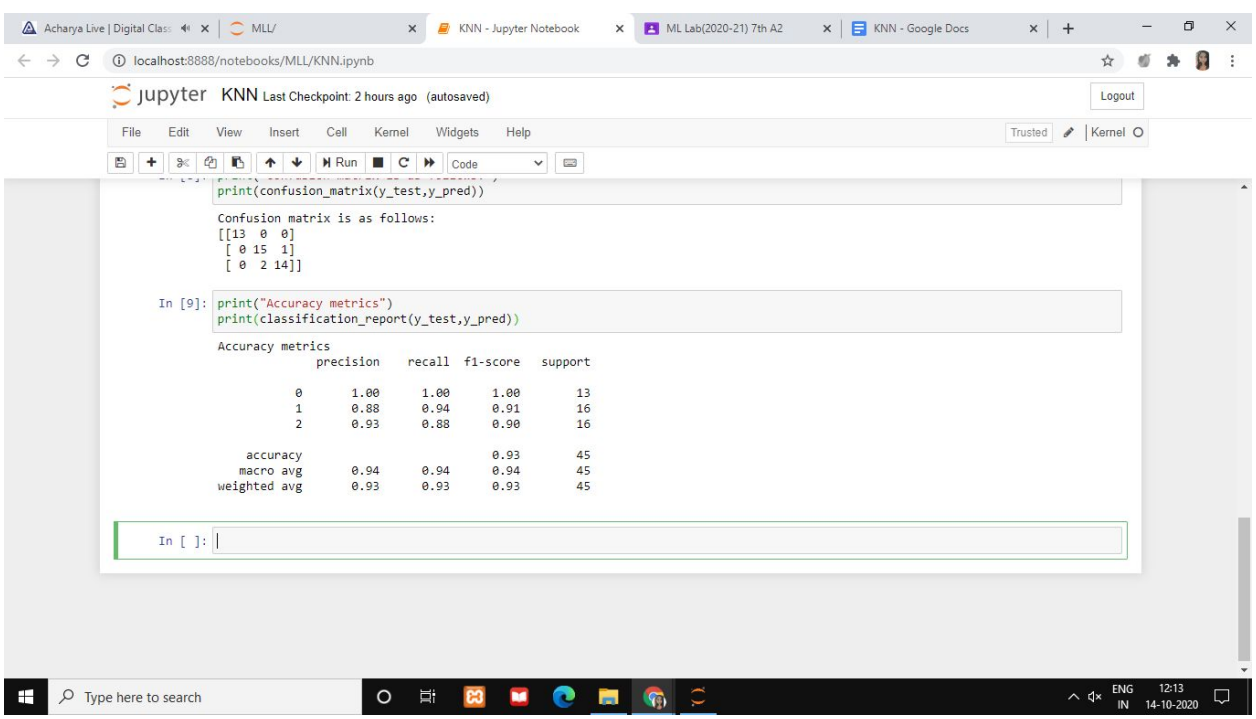
~~Label 0 →~~ $= \frac{13+15+14}{13+15+14+1+2}$

$= \frac{42}{45} = 0.93$


Acharya Live | Digital Class
MLL/
KNN - Jupyter Notebook
ML Lab(2020-21) 7th A2
KNN - Google Docs
localhost:8888/notebooks/MLL/KNN.ipynb
jupyter KNN Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Kernel
Run
Code
In [1]:
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,confusion_matrix
from sklearn import datasets
In [2]:
iris=datasets.load_iris()
print(iris)
{'data': array([[5.1, 3.5, 1.4, 0.2],
[4.9, 3. , 1.4, 0.2],
[4.7, 3.2, 1.3, 0.2],
[4.6, 3.1, 1.5, 0.2],
[5. , 3.6, 1.4, 0.2],
[5.4, 3.9, 1.7, 0.4],
[4.6, 3.4, 1.4, 0.3],
[5. , 3.4, 1.5, 0.2],
[4.4, 2.9, 1.4, 0.2],
[4.9, 3.1, 1.5, 0.1],
[5.4, 3.7, 1.5, 0.2],
[4.8, 3.4, 1.6, 0.2],
[4.8, 3. , 1.4, 0.1],
[4.3, 3. , 1.1, 0.1],
[5.8, 4. , 1.2, 0.2],
[5.7, 4.4, 1.5, 0.4],
[5.4, 3.9, 1.3, 0.4],
[5.1, 3.5, 1.4, 0.3],
[5.7, 3.8, 1.7, 0.3],
In [3]: iris_data=iris.data
Type here to search
ENG
IN
12:13
14-10-2020


Acharya Live | Digital Class
MLL/
KNN - Jupyter Notebook
ML Lab(2020-21) 7th A2
KNN - Google Docs
localhost:8888/notebooks/MLL/KNN.ipynb
jupyter KNN Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Kernel
Run
Code
In [3]:
iris_data=iris.data
iris_labels=iris.target
print(iris_data)
[[5.1 3.5 1.4 0.2]
[4.9 3. 1.4 0.2]
[4.7 3.2 1.3 0.2]
[4.6 3.1 1.5 0.2]
[5. 3.6 1.4 0.2]
[5.4 3.9 1.7 0.4]
[4.6 3.4 1.4 0.3]
[5. 3.4 1.5 0.2]
[4.4 2.9 1.4 0.2]
[4.9 3.1 1.5 0.1]
[5.4 3.7 1.5 0.2]
[4.8 3.4 1.6 0.2]
[4.8 3. 1.4 0.1]
[4.3 3. 1.1 0.1]
[5.8 4. 1.2 0.2]
[5.7 4.4 1.5 0.4]
[5.4 3.9 1.3 0.4]
[5.1 3.5 1.4 0.3]
[5.7 3.8 1.7 0.3]
In [4]: print(iris_labels)
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
Type here to search
ENG
IN
12:13
14-10-2020


Acharya Live | Digital Class
MLL/
KNN - Jupyter Notebook
ML Lab(2020-21) 7th A2
KNN - Google Docs
localhost:8888/notebooks/MLL/KNN.ipynb
jupyter KNN Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Kernel
Run Code
[4.9 3.1 1.5 0.2]
In [4]: print(iris_labels)
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2]
In [5]: x_train,x_test,y_train,y_test=train_test_split(iris_data,iris_labels,test_size=0.30)
In [6]: classifier=KNeighborsClassifier(n_neighbors=5)
classifier.fit(x_train,y_train)
y_pred=classifier.predict(x_test)
In [8]: print('Confusion matrix is as follows:')
print(confusion_matrix(y_test,y_pred))
Confusion matrix is as follows:
[[13 0 0]
[0 15 1]
[0 2 14]]
In [9]: print("Accuracy metrics")
print(classification_report(y_test,y_pred))
Accuracy metrics
precision recall f1-score support
0 1.00 1.00 1.00 13
Type here to search
ENG
IN
12:13
14-10-2020


Acharya Live | Digital Class
MLL/
KNN - Jupyter Notebook
ML Lab(2020-21) 7th A2
KNN - Google Docs
localhost:8888/notebooks/MLL/KNN.ipynb
jupyter KNN Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Kernel
Run Code
print(confusion_matrix(y_test,y_pred))
Confusion matrix is as follows:
[[13 0 0]
[0 15 1]
[0 2 14]]
In [9]: print("Accuracy metrics")
print(classification_report(y_test,y_pred))
Accuracy metrics
precision recall f1-score support
0 1.00 1.00 1.00 13
1 0.88 0.94 0.91 16
2 0.93 0.88 0.90 16
accuracy 0.93 45
macro avg 0.94 0.94 0.94 45
weighted avg 0.93 0.93 0.93 45
In [ ]:
Type here to search
ENG
IN
12:13
14-10-2020