1. Using the iris data set implement the KNN algorithm. Take different values for Test and training data set .Also use different values for k. Also find the accuracy level.

**CODE**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read\_csv("iris.csv")

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 4].values

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors=5)

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

from sklearn.metrics import classification\_report, confusion\_matrix

print(classification\_report(y\_test, y\_pred))

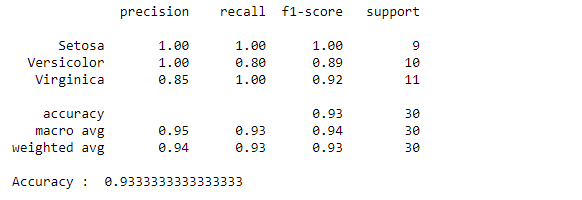
from sklearn.metrics import accuracy\_score

print ("Accuracy : ", accuracy\_score(y\_test, y\_pred))

df = pd.DataFrame({'Real Values':y\_test, 'Predicted Values':y\_pred})

df

**OUTPUT**



Reference: <https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/>

1. Download another data set suitable for the KNN and implement the KNN algorithm. Take different values for Test and training data set .Also use different values for k.

**CODE**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read\_csv("cancer.csv")

dataset.head()

dataset.info()

X = dataset.iloc[:, 2:35].values

print(X)

y = dataset.iloc[:, 1].values

print(y)

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors=5)

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

from sklearn.metrics import classification\_report, confusion\_matrix

print(classification\_report(y\_test, y\_pred))

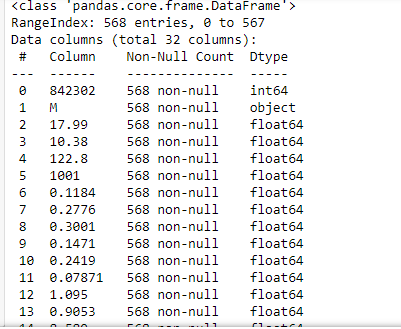
from sklearn.metrics import accuracy\_score

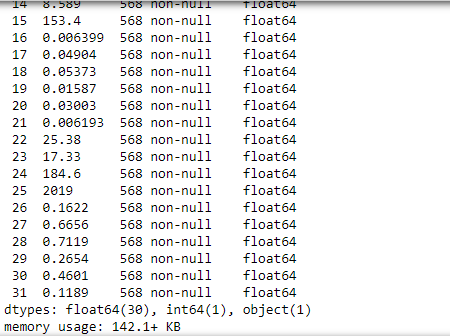
print ("Accuracy : ", accuracy\_score(y\_test, y\_pred))

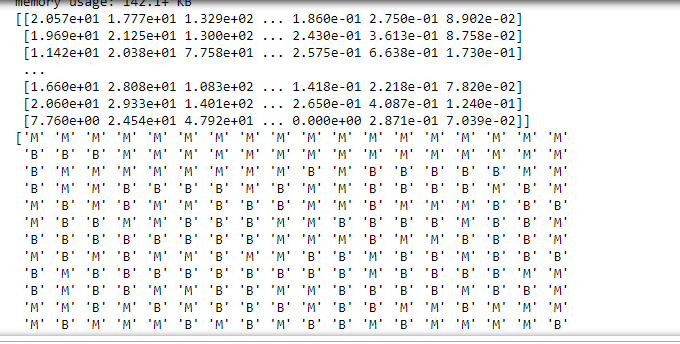
df = pd.DataFrame({'Real Values':y\_test, 'Predicted Values':y\_pred})

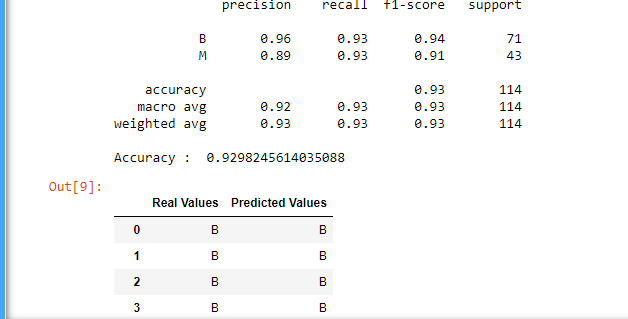
df

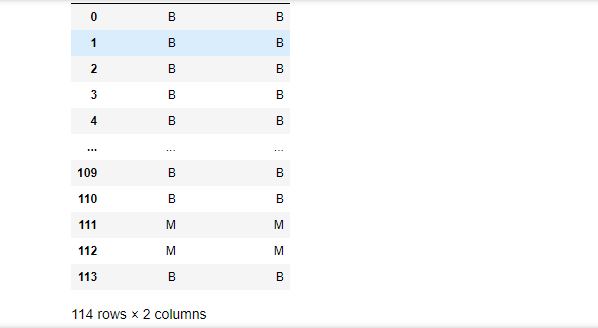
**OUTPUT**











1. Using iris data set, implement naive bayes classification for different naive Bayes classification algorithms.( (i) gaussian (ii) bernoulli etc)

* Find out the accuracy level w.r.t to each  algorithm
* Display the no:of mislabeled classification from test data set
* List out the class labels of the mismatching records

**i)CODE**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read\_csv('iris.csv')

X = dataset.iloc[:,:4].values

y = dataset['variety'].values

dataset.head(5)

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)

from sklearn.naive\_bayes import GaussianNB

classifier = GaussianNB()

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

y\_pred

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

from sklearn.metrics import accuracy\_score

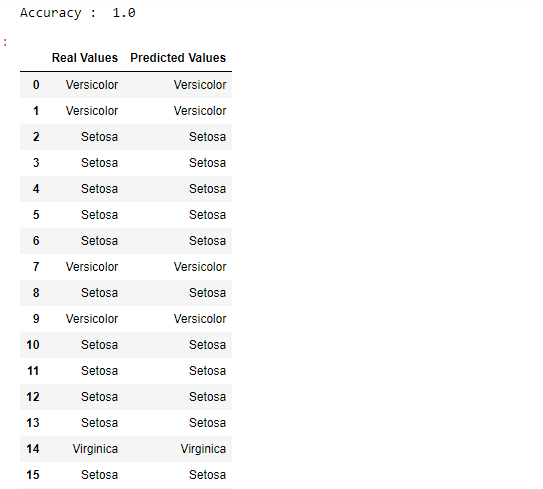
print ("Accuracy : ", accuracy\_score(y\_test, y\_pred))

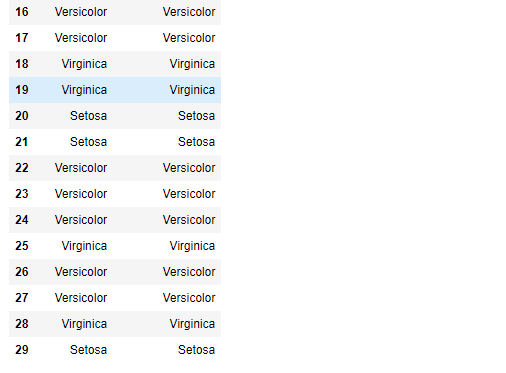
cm

df = pd.DataFrame({'Real Values':y\_test, 'Predicted Values':y\_pred})

df

**OUTPUT**





**CODE**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read\_csv('iris.csv')

X = dataset.iloc[:,:4].values

y = dataset['variety'].values

dataset.head(5)

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)

from sklearn.naive\_bayes import BernoulliNB

classifier = BernoulliNB()

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

y\_pred

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

from sklearn.metrics import accuracy\_score

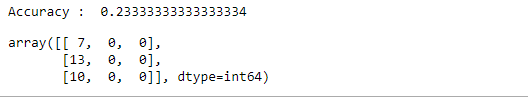
print ("Accuracy : ", accuracy\_score(y\_test, y\_pred))

cm

# df = pd.DataFrame({'Real Values':y\_test, 'Predicted Values':y\_pred})

# df

**OUTPUT**



**ii)CODE**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read\_csv('iris.csv')

X = dataset.iloc[:,:4].values

y = dataset['variety'].values

dataset.head(5)

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)

from sklearn.naive\_bayes import BernoulliNB

classifier = BernoulliNB()

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

y\_pred

#from sklearn.metrics import confusion\_matrix

#cm = confusion\_matrix(y\_test, y\_pred)

from sklearn.metrics import accuracy\_score

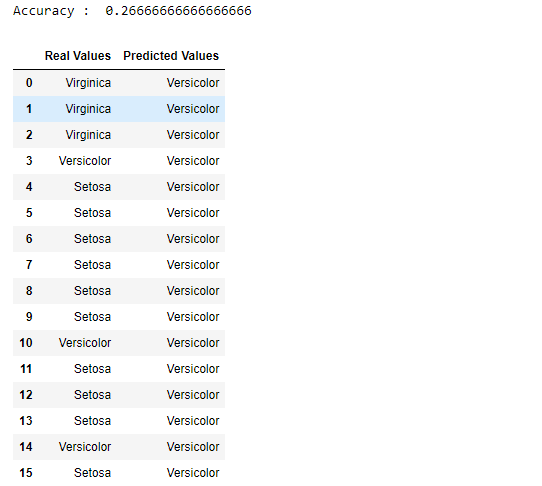
print ("Accuracy : ", accuracy\_score(y\_test, y\_pred))

#cm

df = pd.DataFrame({'Real Values':y\_test, 'Predicted Values':y\_pred})

df

**OUTPUT**





References:

<https://towardsdatascience.com/machine-learning-basics-naive-bayes-classification-964af6f2a965>

<https://scikit-learn.org/stable/modules/classes.html#module-sklearn.naive_bayes>

1. Use car details CSV file and implement decision tree algorithm

* Find out the accuracy level.
* Display the no:of mislabeled classification from test data set
* List out the class labels of the mismatching records

References:

<https://www.24tutorials.com/machine-learning/case-study-decision-tree-model-for-car-quality/>

<https://notebook.community/bMzi/ML_in_Finance/0210_DecisionTrees>

<https://stackabuse.com/decision-trees-in-python-with-scikit-learn/>

For Data Sets Refer:

<https://www.kaggle.com> ( for data set)

<http://archive.ics.uci.edu/ml/datasets.php>