

Harsh Seksaria

2048011

2 - MDS

Machine Learning Lab 10

Naive Bayes Classifier

15 April, 2021

CHRIST (Deemed to be University)

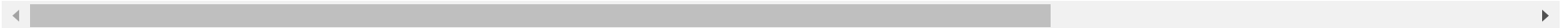
```
In [17]: import pandas as pd
```

```
In [18]: from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer()
fl = pd.DataFrame(cancer['data'], columns=cancer['feature_names'])
fl.head()
```

Out[18]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	25.38	17.33	184.60	2019.0
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	24.99	23.41	158.80	1956.0
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	152.50	1709.0
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	98.87	567.7
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	22.54	16.67	152.20	1575.0

5 rows × 30 columns



```
In [19]: x = fl
y = pd.DataFrame(cancer['target'], columns = ['CancerType'])
x.shape, y.shape
```

Out[19]: ((569, 30), (569, 1))

```
In [20]: #Dividing dataset into training and testing set
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size=0.25, random_state=11)
```

Gaussian NB

```
In [21]: from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, Y_train)

y_pred = gnb.predict(X_test)

from sklearn import metrics
print("Model accuracy:", metrics.accuracy_score(Y_test, y_pred)*100)
```

Model accuracy: 95.1048951048951

D:\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
return f(**kwargs)

Multinomial NB

```
In [22]: from sklearn.naive_bayes import MultinomialNB
gnb = MultinomialNB()
gnb.fit(X_train, Y_train)

y_pred = gnb.predict(X_test)

from sklearn import metrics
print("Model accuracy:", metrics.accuracy_score(Y_test, y_pred)*100)
```

Model accuracy: 90.9090909090909

D:\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
return f(**kwargs)

Bernoulli NB

```
In [23]: from sklearn.naive_bayes import BernoulliNB
gnb = BernoulliNB()
gnb.fit(X_train, Y_train)

y_pred = gnb.predict(X_test)

from sklearn import metrics
print("Model accuracy:", metrics.accuracy_score(Y_test, y_pred)*100)
```

Model accuracy: 65.03496503496503

D:\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```
return f(**kwargs)
```

Conclusion

We can clearly see that the Gaussian NB algorithm gave the highest accuracy among the other algorithms, which is above 95%.