Manas Jha

RA1911003010643

Aim- Implementation of learning algorithms for an application

Problem Formulation- Solving a dataset using machine learning algorithm.

Problem Statement-

The dataset belongs to classic UCI Machine Learning Repository

Given:

1. Brest Cancer Dataset
2. Features related to the Breast Cancer
3. Aim is to predict whether the Tumor is Benign or Malignant
4. Divided into two classes where 2 - Benign and 4 – Malignant

Method:

Training the dataset with different Machine Learning models and concluding which Model Gives the Highest Accuracy.

Algorithm-

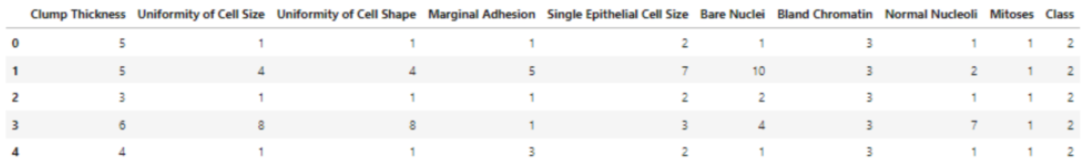
Importing Libraries

- Data Preprocessing
- Splitting Data into test set and training set
- Feature Scaling
- Training data in Random forest classification
- Predict for a single value
- Dropping the Sample Code Number as it has no influence over the Classification
- it also can Reduce the Accuracy of the model
- X is the Independent Variable and Y is the Dependent Variable
- Printing the Confusion Matrix
- Now that we know Random Forest algorithm gives highest accuracy, trying to predict the class.
- The class is predicted according to the values of the respective features
- Therefore it has predicted that for these set of feature values the tumor is going to be benign i.e. class2

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('Breast Cancer Dataset.csv')
dataset.drop('Sample code number',
  axis='columns', inplace=True)
#Dropping the Sample Code Number as it has no influence over the Classification
#it also can Reduce the Accuracy of the model

X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
#X is the Independent Variable and Y is the Dependent Variable
dataset.head()

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.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)

X_test = sc.transform(X_test)


from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
classifier_forest = RandomForestClassifier(n_estimators = 10, criterion =
'entropy')
classifier_forest.fit(X_train, y_train)
y_pred = classifier_forest.predict(X_test)
cm_forest = confusion_matrix(y_test, y_pred)
print(cm_forest)
acc_score_forest = accuracy_score(y_test, y_pred)
##Printing the Confusion Matrix




#To Easily know which has highest accuracy
```

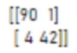
```
x = ["Naive Bayes", "Decision Tree", "Logistic Regression", "K_NN","Random
Forest","SVM","Kernal SVM"]
y = [100*acc_score_bayes,100*acc_score_tree, 100*acc_score_log_reg,
100*acc_score_knn,100*acc_score_forest,100*acc_score_SVM_lin,100*acc_score_SVM_
rbf]
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])

plt.barh(x, y)

for index, value in enumerate(y):
    plt.text(value, index,
             str(value))

plt.show()


#Now that we know Random Forest algorithm gives highest accuracy , trying to
predict the class .
#The class is predicted according to the values of the repective features
print(classifier_forest.predict(sc.transform([[4,8,1,2,2,5,3,2,1]])))
#Therefore it has predicted that for these set of feature values the tumor is
going to be Benign i.e class 2
```

OUTPUT:



	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
0	5	1	1	1	2	1	3	1	1	2
1	5	4	4	5	7	10	3	2	1	2
2	3	1	1	1	2	2	3	1	1	2
3	6	8	8	1	3	4	3	7	1	2
4	4	1	1	3	2	1	3	1	1	2

CONFUSION MATRIX:



```
[[90  1]
 [ 4 42]]
```

Observation:

- From the above Accuracy Scores Random Forest Classifier ML model has given the highest Accuracy
- Observing the Confusion Matrix
    - ♣ Out of 91 Dependent Values (class) only 1 value was predicted wrong and 90 values were Predicted correctly which in turn gave the high accuracy.

Result: Hence a Classification algorithm was implemented.