ASSIGNMENT 4 : Design and implement the following classifiers
DATE : *17-10-2020* **SUBMITTED BY** : *DWDM20G05*

OBJECTIVE :

- Design and implement the following classifiers.
  - Regression classifier.
  - Naïve Bayesian Classifier.
  - k-NN classifier (Take k = 1,3,5,7)
  - Three layer Artificial Neural Network (ANN) classifier (use back propagation).
- Use IRIS, Breast Cancer Wisconsin data sets from UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/).
- Extract different training and test data sets from the original data set.
- Tabulate the results with classification accuracy.
- Construct confusion matrix for each reading of training/test data sets.

INTRODUCTION :

Classification is the process of predicting the class of given data points. Classes are sometimes called targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).
For example, spam detection in email service providers can be identified as a classification problem. This is s binary classification since there are only 2 classes as spam and not spam. A classifier utilizes some training data to understand how given input variables relate to the class. In this case, known spam and non-spam emails have to be used as the training data. When the classifier is trained accurately, it can be used to detect an unknown email.

THEORY :

Regression Classifier: Simple linear regression is an approach for predicting a response using a single feature. It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).

Y (predicted) = m(gradient) x + c(intercept)

Naive Bayes Classifiers: Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

k-NN Algorithm: K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

Artificial Neural Network: Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called 'edges'. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

OBSERVATIONS :

Activity 1 : Logistic Regression
Iris Dataset

```
for iris dataset
training accuracy:
0.9555555555555556
validation  accuracy:
0.9333333333333333
testing  accuracy:
1.0
confusion matrix
[[18  0  0]
 [ 0  6  0]
 [ 0  0  6]]
```

Breast-Cancer Dataset

```
training accuracy:
0.9002932551319648
validation  accuracy:
0.9298245614035088
testing  accuracy:
0.9210526315789473
confusion matrix
[[24  8]
 [ 1 81]]
```

Activity 2 : KNN

For k=1:

Iris:

```
for Iris data:
Train set: 100
Test set: 50
k: 1

Accuracy: 82.0%
printing confusion matrix for KNN Classifier
[[12  0  0]
 [ 0 15  1]
 [ 0  8 14]]
for iris data:
Train set: 96
Test set: 54
k: 1

Accuracy: 94.44444444444444%
printing confusion matrix for KNN Classifier
[[17  0  0]
 [ 0 14  1]
 [ 0  2 20]]
```

Breast Cancer:

```
for Breast Cancer data:
Train set: 459
Test set: 239
k: 1

Accuracy: 57.32217573221757%
printing confusion matrix for KNN Classifier
[[103  60]
 [ 42  34]]
for Breast Cancer data:
Train set: 454
Test set: 244
k: 1

Accuracy: 59.01639344262295%
printing confusion matrix for KNN Classifier
[[118  36]
 [ 64  26]]
```

For k=3:

Iris

```
for Iris data:
Train set: 103
Test set: 47
k: 3

Accuracy: 95.74468085106383%
printing confusion matrix for KNN Classifier
[[16  0  0]
 [ 0 16  0]
 [ 0  2 13]]
for iris data:
Train set: 96
Test set: 54
k: 3

Accuracy: 92.5925925925926%
printing confusion matrix for KNN Classifier
[[17  0  0]
 [ 0 17  2]
 [ 0  2 16]]
```

Breast Cancer:

```
for Breast Cancer data:
Train set: 490
Test set: 208
k: 3

Accuracy: 59.61538461538461%
printing confusion matrix for KNN Classifier
[[98 38]
 [46 26]]
for Breast Cancer data:
Train set: 477
Test set: 221
k: 3

Accuracy: 59.276018099547514%
printing confusion matrix for KNN Classifier
[[107  36]
 [ 54  24]]
```

For k= 5:
Iris dataset

```
for Iris data:
Train set: 95
Test set: 55
k: 5

Accuracy: 89.0909090909091%
printing confusion matrix for KNN Classifier
[[15  0  0]
 [ 0 21  0]
 [ 0  6 13]]
for iris data:
Train set: 93
Test set: 57
k: 5

Accuracy: 92.98245614035088%
printing confusion matrix for KNN Classifier
[[20  0  0]
 [ 0 19  1]
 [ 0  3 14]]
```

Breast Cancer:

```
for Breast Cancer data:
Train set: 470
Test set: 228
k: 5

Accuracy: 61.8421052631579%
printing confusion matrix for KNN Classifier
[[116  21]
 [ 66  25]]
for Breast Cancer data:
Train set: 493
Test set: 205
k: 5

Accuracy: 64.39024390243902%
printing confusion matrix for KNN Classifier
[[112  27]
 [ 46  20]]
```

For k = 7:

Iris dataset

```
D:\Academics\Final Year\DWDM\coll\Assignment-4\KNN>python KNN.py
for Iris data:
Train set: 106
Test set: 44
k: 7

Accuracy: 95.45454545454545%
printing confusion matrix for KNN Classifier
[[17  0  0]
 [ 0 13  0]
 [ 0  2 12]]
for iris data:
Train set: 100
Test set: 50
k: 7

Accuracy: 88.0%
printing confusion matrix for KNN Classifier
[[12  0  0]
 [ 0 16  1]
 [ 0  5 16]]
```

Breast-Cancer

```
D:\Academics\Final Year\DWDM\coll\ass4-1\KNN>python KNN.py
for Cancer data:
Train set: 494
Test set: 204
k: 7

Accuracy: 64.2156862745098%
printing confusion matrix for KNN Classifier
[[108  28]
 [ 45  23]]
for cancer data:
Train set: 484
Test set: 214
k: 7

Accuracy: 66.82242990654206%
printing confusion matrix for KNN Classifier
[[119  20]
 [ 51  24]]
```

*Refer File for naiye-bayes observations.

CONCLUSION :

We have successfully Analysed different classifiers for iris and Breast Cancer dataset and derived accuracies, confusion matrix