Harsh Seksaria

2048011

2 - MDS

Machine Learning Lab - 5 and 6

ANN, SVM and Logistic Regression

18 March, 2021

CHRIST (Deemed to be University)

In [2]:
```
#Import statements
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:
```
#ML Libraries
from sklearn.model_selection import train_test_split
```

Dataset

In [4]:
```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

# Assign colum names to the dataset
colnames = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']

# Read dataset in dataframe
data = pd.read_csv(url, names=colnames)
```

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
sepal-length    0
sepal-width     0
petal-length    0
petal-width     0
Class           0
dtype: int64
```

There are no null values in the dataset.

In [6]:
```
#Dividing dataset into x and y
x = data.drop('Class', axis=1)
y = data['Class']
```

In [7]:
```
#Dividing dataset into training and testing set
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size=0.25, random_state=11)
```

Artificial Neural Networks

In [23]:
```
%%time
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(10, 10, 10), max_iter=1000)
mlp.fit(X_train, Y_train.values.ravel())
```

```
Wall time: 528 ms
```

Out[23]:
```
MLPClassifier(hidden_layer_sizes=(10, 10, 10), max_iter=1000)
```

Prediction

In [19]:
```
Y_pred_ann = mlp.predict(X_test)
```

In [20]:
```
#Printing the predictions
Y_pred_ann
```

Out[20]:
```
array(['Iris-virginica', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-versicolor'],
      dtype='<U15')
```

Evaluation

In [21]:
```
from sklearn.metrics import classification_report, confusion_matrix
print("CLASSIFICATION REPORT:\n\n", classification_report(Y_test, Y_pred_ann))
print("\nCONFUSION MATRIX:\n\n", confusion_matrix(Y_test, Y_pred_ann))
```

```
CLASSIFICATION REPORT:

                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        12
Iris-versicolor       0.92      1.00      0.96        11
 Iris-virginica       1.00      0.93      0.97        15

       accuracy                           0.97        38
      macro avg       0.97      0.98      0.97        38
   weighted avg       0.98      0.97      0.97        38


CONFUSION MATRIX:

 [[12  0  0]
 [ 0 11  0]
 [ 0  1 14]]
```

Advantages of ANN

1. Ability to work with incomplete knowledge
2. Information such as in traditional programming is stored on the entire network, not on a database
3. Fault tolerance- Eeen if one node fails, others keep working successfully
4. Parallel Processing
5. Having a distributed memory

Disadvantages of ANN

1. Hardware dependence
2. Unexplained behaviour of the network
3. Execution duration
4. Difficulty of showing the problem to the network

Support Vector Machine

Model - Gaussian Kernel

In [8]:
```
%%time
from sklearn.svm import SVC
svcclassifier = SVC(kernel="rbf")
svcclassifier.fit(X_train, Y_train)
```

```
Wall time: 54.4 ms
```

Out[8]:
```
SVC()
```

Prediction

In [9]:
```
Y_pred_svm = svcclassifier.predict(X_test)
```

In [10]:
```
#Printing the predictions
Y_pred_svm
```

Out[10]:
```
array(['Iris-virginica', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica'], dtype=object)
```

Evaluation

In [11]:
```
from sklearn.metrics import classification_report, confusion_matrix
print("CLASSIFICATION REPORT:\n\n", classification_report(Y_test, Y_pred_svm))
print("\nCONFUSION MATRIX:\n\n", confusion_matrix(Y_test, Y_pred_svm))
```

```
CLASSIFICATION REPORT:

                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        12
Iris-versicolor       0.83      0.91      0.87        11
 Iris-virginica       0.93      0.87      0.90        15

       accuracy                           0.92        38
      macro avg       0.92      0.93      0.92        38
   weighted avg       0.92      0.92      0.92        38


CONFUSION MATRIX:

 [[12  0  0]
 [ 0 10  1]
 [ 0  2 13]]
```

Advantages of SVM

1. SVM works relatively well when there is a clear margin of separation between classes
2. SVM is more effective in high dimensional spaces
3. SVM is effective in cases where the number of dimensions is greater than the number of samples
4. SVM is relatively memory efficient

Disadvantages of SVM

1. SVM algorithm is not suitable for large data sets
2. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping
3. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform
4. As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification

Logistic Regression

Model

In [12]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

In [13]:
```
%%time
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train, Y_train)
```

```
Wall time: 47.9 ms

D:\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning: lbfgs failed to converge (sta
tus=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.h
tml)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modu
les/linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

Out[13]:
```
LogisticRegression()
```

Prediction

In [14]:
```
Y_pred_lr = svcclassifier.predict(X_test)
```

In [15]:
```
#Printing the predictions
Y_pred_lr
```

Out[15]:
```
array(['Iris-virginica', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica'], dtype=object)
```

Evaluation

In [16]:
```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
print("CLASSIFICATION REPORT:\n\n", classification_report(Y_test, Y_pred_lr))
print("\nCONFUSION MATRIX:\n\n", confusion_matrix(Y_test, Y_pred_lr))
print("\nACCURACY SCORE: ", accuracy_score(Y_test, Y_pred_lr))
```

```
CLASSIFICATION REPORT:

                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        12
Iris-versicolor       0.83      0.91      0.87        11
 Iris-virginica       0.93      0.87      0.90        15

       accuracy                           0.92        38
      macro avg       0.92      0.93      0.92        38
   weighted avg       0.92      0.92      0.92        38


CONFUSION MATRIX:

 [[12  0  0]
 [ 0 10  1]
 [ 0  2 13]]

ACCURACY SCORE:  0.9210526315789473
```

Advantages of Logistic Regression

1. Logistic regression is easier to implement, interpret, and very efficient to train
2. It makes no assumptions about distributions of classes in feature space
3. It can easily extend to multiple classes(multinomial regression) and a natural probabilistic view of class predictions
4. It is very fast at classifying unknown records

Disadvantages of Logistic Regression

1. If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting
2. It constructs linear boundaries
3. The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables
4. Non-linear problems can’t be solved with logistic regression because it has a linear decision surface. Linearly separable data is rarely found in real-world scenarios