Name - Arjun A.

Roll number - 181CO109

Date of submission - 19-3-2021

This notebook was written in google colab.
Link to view notebook
https://colab.research.google.com/drive/1mwYxrMaSh79F-_-qWt6lN0ysPnJvCUEt?usp=sharing

ML Lab 9 - Logistic Regression

This notebook is used to implement Logistic Regression to do classification.

Importing necessary packages

```
from sklearn.datasets import load_iris
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

Importing data from the iris dataset

Using the load_iris function from sklearn, importing the iris dataset

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html#sklearn.datasets.load_iris

```
# Function to import data
def importdata():

  irisData = load_iris()
  X = irisData.data
  Y = irisData.target
  names = irisData.target_names
  featureName = irisData.feature_names
  print(type(X))
  print(type(Y))
```

```
  print(names)

  return X, Y, names, featureName
```

```
X, Y, irisClassNames, irisFeatureNames = importdata()
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
['setosa' 'versicolor' 'virginica']
```

Splitting the data into train and test sets

Splitting the data in the ratio of 7:3. (70% training and 30% testing)

```
def splitdataset(X, Y):

  X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.3, random_st

  return X, Y, X_train, X_test, y_train, y_test
```

```
X, Y, X_train, X_test, y_train, y_test = splitdataset(X, Y)

#Use to print the entire dataset
#print(X, Y, X_train, X_test, y_train, y_test, sep = '\n\n')

#Printing size of the split
print('Test dataset size\nX_test -', len(X_test), '\ny_test -', len(y_test), '\n')
print('Train dataset size\nX_train -', len(X_train), '\ny_train -', len(y_train))
```

```
Test dataset size
X_test - 45
y_test - 45

Train dataset size
X_train - 105
y_train - 105
```

Defining model

```
logRegression = LogisticRegression(max_iter= 1000)
```

Training model on train dataset

```
logRegression.fit(X_train,y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=1000,
                   multi_class='auto', n_jobs=None, penalty='l2',
```

```
                        random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                        warm_start=False)
```

Calculating accuracy on the model using the test set

```
1 y_pred = logRegression.predict(X_test)
2 print('Accuracy =',logRegression.score(X_test,y_test), '\n')
3 print('Confusion Matrix\n')
4 print(confusion_matrix(y_test, y_pred))
5 print('\nClassification report\n')
6 print(classification_report(y_test, y_pred))
```

```
Accuracy = 0.9777777777777777

Confusion Matrix

[[16  0  0]
 [ 0 11  0]
 [ 0  1 17]]

Classification report

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       0.92      1.00      0.96        11
           2       1.00      0.94      0.97        18

    accuracy                           0.98        45
   macro avg       0.97      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```