In [19]:
```
# Load Libraries
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

import warnings
warnings.filterwarnings("ignore")

# Load dataset
filename = 'D:\\Dataset\IRIS.csv'
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class'
dataset = read_csv(filename, names=names)

#print(dataset.shape)
#print(dataset.head(20))
#print(dataset.describe())
#print(dataset.groupby('class').size())

# histograms
#dataset.hist()
#pyplot.show()

# Split-out validation dataset
array = dataset.values
X = array[:,0:4]
Y = array[:,4]

# Spot-Check Algorithms
models = []
models.append(('LR', LogisticRegression()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
# evaluate each model in turn
results = []
names = []


"""------------------------- Train test split------------------------------
test_size = 0.20
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=test_size)

for name, model in models:
    model.fit(X_train, Y_train)
    result = model.score(X_test, Y_test)
    print(name, result*100.0)

------------------------------------------------------------------------------
for name, model in models:
    kfold = KFold(n_splits=7)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='a
    results.append(cv_results)
    names.append(name)
    print(name, cv_results.mean()*100)
```

```
LR 95.00466853408031
LDA 98.31932773109244
KNN 95.00466853408031
CART 93.32399626517274
NB 93.32399626517274
SVM 95.00466853408031
```

In [29]:
```
# Load libraries
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

import warnings
warnings.filterwarnings("ignore")

# Load dataset
filename = 'D:\\Dataset\IRIS.csv'
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class'
dataset = read_csv(filename, names=names)

#print(dataset.shape)
#print(dataset.head(20))
#print(dataset.describe())
#print(dataset.groupby('class').size())

# histograms
#dataset.hist()
#pyplot.show()

# Split-out validation dataset
array = dataset.values
X = array[:,0:4]
Y = array[:,4]

# Spot-Check Algorithms
models = LinearDiscriminantAnalysis()
kfold = KFold(n_splits=10)
results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy
print("Training Accuracy:", results.mean()*100)

models.fit(X_train,Y_train)
predictions = models.predict(X_test)
print(confusion_matrix(Y_test, predictions))
print(classification_report(Y_validation, predictions))

#_____________________ Implemenattion_____________________________________
#Predict the output for a specific iinput
model.fit(X,Y)
test = [[5, 2.5, 4.9, 1.6]]
print(model.predict(test))
```

```
Training Accuracy: 95.83333333333333
[[ 9  0  0]
 [ 0 14  0]
 [ 0  0  7]]
                 precision    recall  f1-score   support

    Iris-setosa       0.33      0.38      0.35         8
Iris-versicolor       0.36      0.50      0.42        10
 Iris-virginica       0.57      0.33      0.42        12

       accuracy                           0.40        30
      macro avg       0.42      0.40      0.40        30
   weighted avg       0.44      0.40      0.40        30

['Iris-virginica']
```

In [ ]: