In [1]:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Nov  1 17:44:49 2021

@author: stephaniewatkins
"""

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import os
from scipy.stats import norm
from scipy import stats
import warnings
warnings.filterwarnings('ignore')
from time import time
from IPython.display import display # Allows the use of display() for DataFrames
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn import datasets
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics


seeds=pd.read_table('~/Desktop/DANN862/seeds.csv', sep=',')
seeds = seeds.dropna()
seeds=seeds.drop_duplicates()
seeds.head()
seeds.info()
seeds.describe(include='all')
print(seeds.shape)
print(seeds.isnull().sum())
figsize=plt.rcParams['figure.figsize']
seeds.hist(figsize=(20,16), color='r')
corr = seeds.corr()
# plot correlation matrix
fig = plt.figure(figsize=(7, 5.5))
mask = np.zeros_like(corr, dtype=np.bool) # create mask to cover the upper trian
mask[np.triu_indices_from(mask)] = True
sns.heatmap(corr, annot=True, mask=mask, vmax=0.5,linewidths=0.1)
fig.suptitle('Attribute Correlation Matrix', fontsize=14)

# Dividing Data into train and test
train_data = seeds[['Area', 'Perimeter', 'Compactness', 'Len_Kernel', 'Wid_Kerne
test_data = seeds[['Area', 'Perimeter', 'Compactness', 'Len_Kernel', 'Wid_Kernel
X = seeds.iloc[:,0:7]
y = seeds['Classification']
seeds.columns
from sklearn.cluster import KMeans
from sklearn import metrics
kmeans = KMeans(n_clusters = 3, random_state = 130)
y_pred = kmeans.fit_predict(X)
metrics.homogeneity_score(y,y_pred)
metrics.completeness_score(y,y_pred)
```

```
metrics.adjusted_rand_score(y,y_pred)
metrics.silhouette_score(X,y_pred, metric='euclidean')
centers = kmeans.cluster_centers_
centers_pl = centers[:,1]
centers_pw = centers[:,1]
plt.figure(figsize=(12,6))
plt.subplot(121)
plt.scatter(X.Area, X.Perimeter, c=y)
plt.title('Seed Data')
plt.xlabel('Area')
plt.ylabel('Perimeter')
plt.subplot(122)
plt.scatter(X.Area, X.Perimeter, c=y_pred, label = 'Predicted')
plt.scatter(centers_pl, centers_pw, s=100, c='r', marker = 'x', label = 'Cluster
plt.title('Clustering results')
plt.xlabel('Area')
plt.ylabel('Perimeter')
plt.legend()
plt.subplots_adjust(wspace=0.2)
#Average Linkage Type
from sklearn.cluster import AgglomerativeClustering
Hier_ypred1 = AgglomerativeClustering(n_clusters = 3, affinity='euclidean',linka
metrics.homogeneity_score(y,Hier_ypred1)
metrics.completeness_score(y,Hier_ypred1)
metrics.adjusted_rand_score(y,Hier_ypred1)
metrics.silhouette_score(X, Hier_ypred1, metric='euclidean')
from scipy.cluster.hierarchy import dendrogram, linkage
Zavg = linkage(X,method = 'average')
plt.figure(figsize=(100,100))
den=dendrogram(Zavg, leaf_font_size=8)
#Complete Linkage Type
from sklearn.cluster import AgglomerativeClustering
Hier_ypred2 = AgglomerativeClustering(n_clusters = 3, affinity = 'euclidean', li
metrics.homogeneity_score(y, Hier_ypred2)
metrics.completeness_score(y,Hier_ypred2)
metrics.silhouette_score(X,Hier_ypred2,metric = 'euclidean')
#ward linkkage
from sklearn.cluster import AgglomerativeClustering
Hier_ypred3 = AgglomerativeClustering(n_clusters=3, affinity = 'euclidean',linka
metrics.homogeneity_score(y,Hier_ypred3)
metrics.completeness_score(y,Hier_ypred3)
metrics.adjusted_rand_score(y,Hier_ypred3)
metrics.silhouette_score(X, Hier_ypred3, metric = 'euclidean')

#Mean or average linkage clustering:
#average of dissimilarities is the distance  between two clusters
print("The mean/average linkage clustering results are below")
print("the homogeneity score is ", metrics.homogeneity_score(y,Hier_ypred1))
print("thecompleteness score score is ", metrics.completeness_score(y,Hier_ypred
print("the adjusted random score score is ", metrics.adjusted_rand_score(y,Hier_
print("the silhoutte score is ", metrics.silhouette_score(X, Hier_ypred1, metric
#max or complete linkage computes all pairwise dissimilarties between elenents i
#considers largest value (max) of dissimilarites and tends to produce more compa
print("The maximum/complete linkage clustering results are below")
print("the homogeneity score is ", metrics.homogeneity_score(y,Hier_ypred2))
print("thecompleteness score score is ", metrics.completeness_score(y,Hier_ypred
print("the adjusted random score score is ", metrics.adjusted_rand_score(y,Hier_
print("the silhoutte score is ", metrics.silhouette_score(X, Hier_ypred2, metric

#wards min variance minimizes total within the cluster variance
#at each step the pair of clusters with min between cluster distance are merged
```

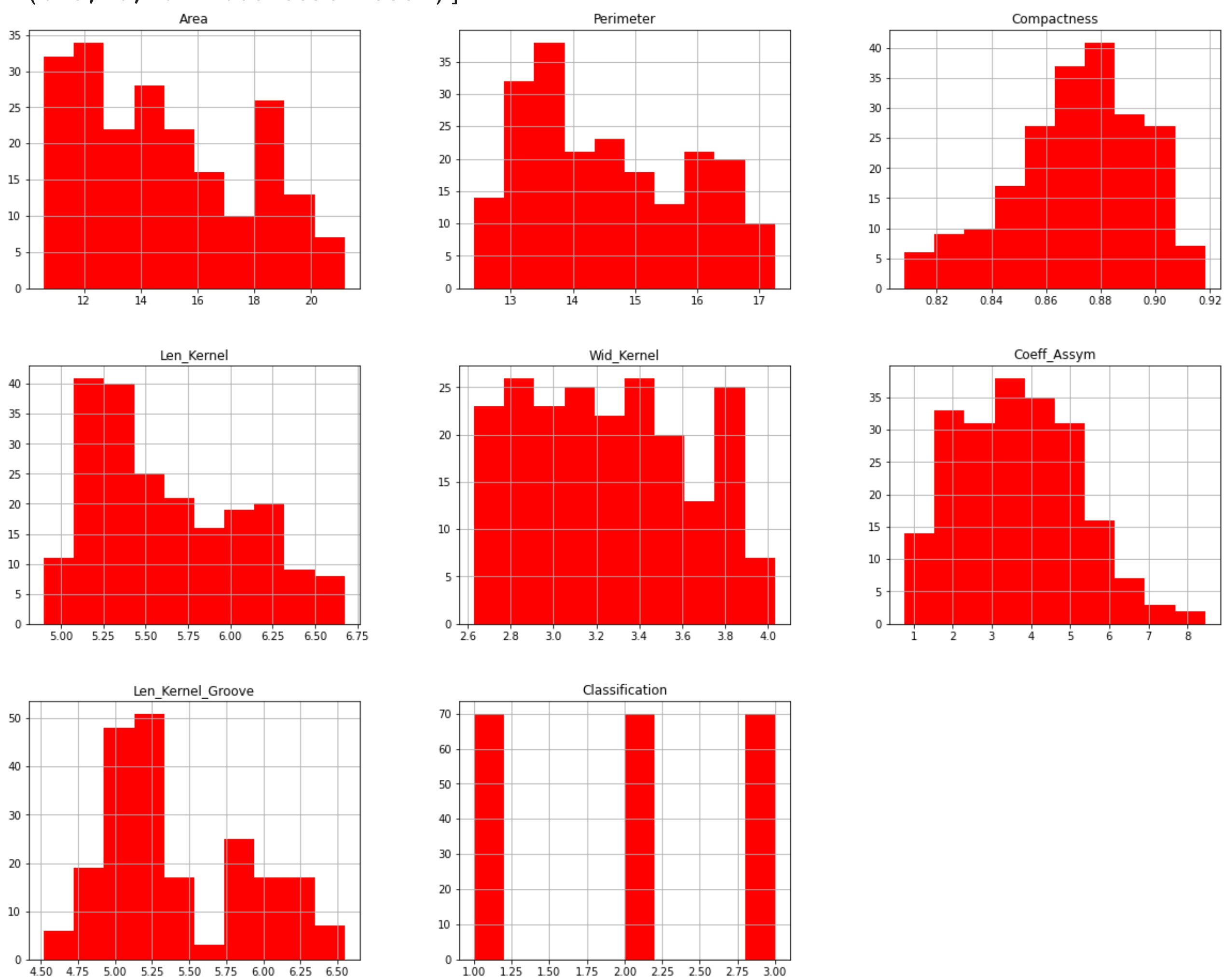
```
print("The wards min variance linkage clustering results are below")
print("the homogeneity score is ", metrics.homogeneity_score(y,Hier_ypred3))
print("thecompleteness score score is ", metrics.completeness_score(y,Hier_ypred
print("the adjusted random score score is ", metrics.adjusted_rand_score(y,Hier_
print("the silhoutte score is ", metrics.silhouette_score(X, Hier_ypred3, metric
#based on the overall scoring the mean/average linkage clustering performed the

#4
from sklearn.cluster import DBSCAN
from sklearn import datasets
from sklearn.cluster import KMeans
import numpy as np
X.head()
y.head()
metrics.homogeneity_score(y, y_pred)
metrics.completeness_score(y, y_pred)
metrics.adjusted_rand_score(y,y_pred)
metrics.silhouette_score(X, y_pred, metric = 'euclidean')
result = []
epses = [0.4, 0.6, 0.8]
min_samples=[5,10,15]
range_n_clusters = [2, 3, 4, 5, 6]
for v in epses:
    for n in range_n_clusters:
        y_pred_temp = DBSCAN(eps = v, min_samples = n).fit_predict(X)
        score = metrics.silhouette_score(X,y_pred_temp, metric = 'euclidean')
        result.append((v,n,score))
result
#  (0.8, 6, 0.2266848593143534)] was the best value
```

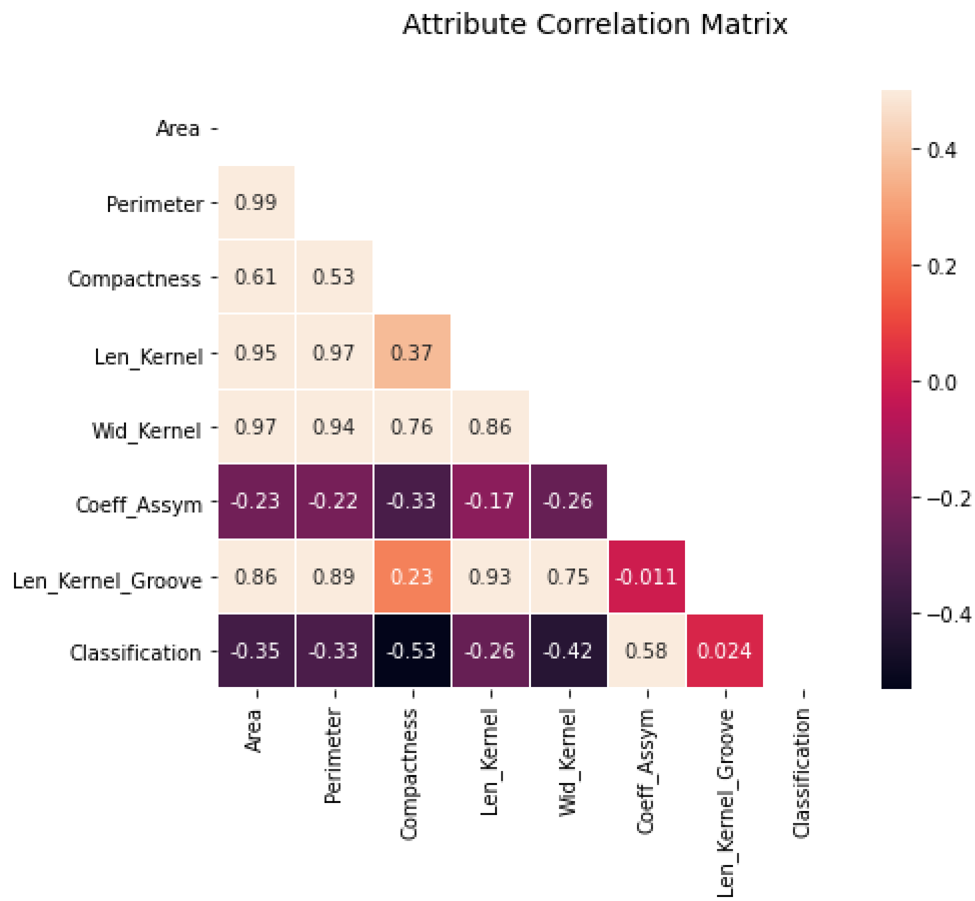
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 210 entries, 0 to 209
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Area               210 non-null    float64
 1   Perimeter          210 non-null    float64
 2   Compactness        210 non-null    float64
 3   Len_Kernel         210 non-null    float64
 4   Wid_Kernel         210 non-null    float64
 5   Coeff_Assym        210 non-null    float64
 6   Len_Kernel_Groove  210 non-null    float64
 7   Classification     210 non-null    int64  
dtypes: float64(7), int64(1)
memory usage: 14.8 KB
(210, 8)
Area                 0
Perimeter            0
Compactness          0
Len_Kernel           0
Wid_Kernel           0
Coeff_Assym          0
Len_Kernel_Groove    0
Classification       0
dtype: int64
The mean/average linkage clustering results are below
the homogeneity score is  0.7131289164537258
thecompleteness score score is  0.7170764841647859
the adjusted random score score is  0.7441752360248661
the silhoutte score is  0.4581123750095183
The maximum/complete linkage clustering results are below
the homogeneity score is  0.6063655325391804
thecompleteness score score is  0.6241956655849628
```

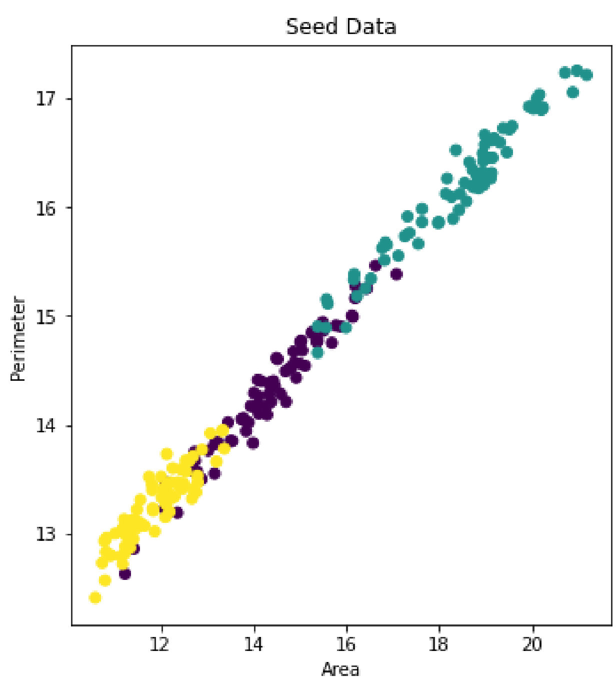
```
the adjusted random score score is  0.546135027762822
the silhoutte score is  0.44140259950708405
The wards min variance linkage clustering results are below
the homogeneity score is  0.7266916689197686
thecompleteness score score is  0.7352022993399162
the adjusted random score score is  0.7131537289031059
the silhoutte score is  0.449360862303228
```

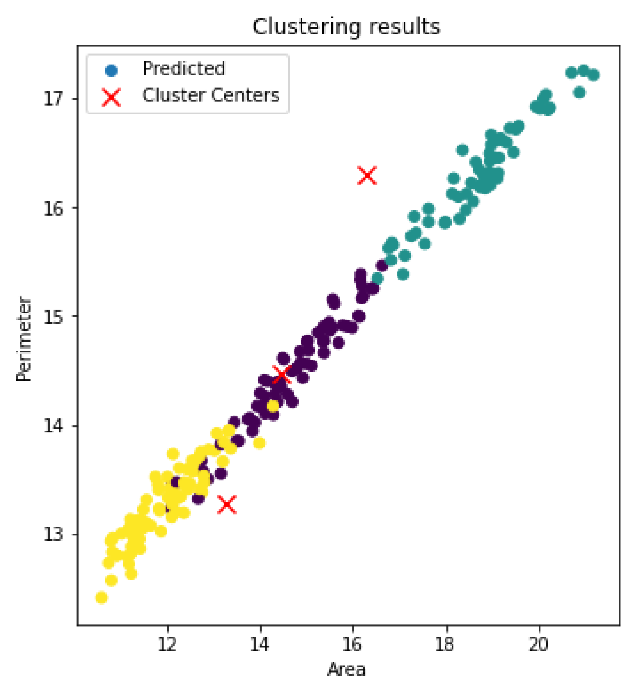
Out[1]:

```
[(0.4, 2, -0.0739148564893898),
 (0.4, 3, -0.2451060513690738),
 (0.4, 4, -0.3320462245762074),
 (0.4, 5, -0.4022809692115891),
 (0.4, 6, -0.3153897088427905),
 (0.6, 2, 0.09827235572852824),
 (0.6, 3, 0.0985372127907852),
 (0.6, 4, 0.03188959052400087),
 (0.6, 5, -0.01993054557130566),
 (0.6, 6, 0.019902030140684042),
 (0.8, 2, -0.052948043226613555),
 (0.8, 3, 0.10288855879545815),
 (0.8, 4, 0.07007579674193141),
 (0.8, 5, 0.2501384260001313),
 (0.8, 6, 0.2266848593143534)]
```




Attribute Correlation Matrix
Area
Perimeter
Compactness
Len_Kernel
Wid_Kernel
Coeff_Assym
Len_Kernel_Groove
Classification
0.99
0.61 0.53
0.95 0.97 0.37
0.97 0.94 0.76 0.86
-0.23 -0.22 -0.33 -0.17 -0.26
0.86 0.89 0.23 0.93 0.75 -0.011
-0.35 -0.33 -0.53 -0.26 -0.42 0.58 0.024
0.4
0.2
0.0
−0.2
−0.4


Seed Data
Perimeter
Area
17
16
15
14
13
12
14
16
18
20


Clustering results
Predicted
Cluster Centers
Perimeter
Area
17
16
15
14
13
12
14
16
18
20