Phase2

February 15, 2024

[ ]:

```python
###
# This code file contains an import of all of the libraries we expect to use in the final project at the top.
# Unused imports will be removed in the phase 4 submission.
# Then the dataset is loaded in as a CSV file downloaded from the same directory.
# We summarize the dataset and replace the Class names with 0 and 1 before starting the cleaning process we check for missing values.
# This particular dataset had no missing values so the only step of the cleaning process that changes the dataset is removal of outliers.
# We include data visualizations before and after removing the outliers based off a z score of 3.
###
```

[ ]:

```python
# Load libraries
# remove unused imports in final project if code is turned in
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
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.linear_model import LinearRegression
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 sklearn as sk
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures
from scipy import stats
import sklearn.metrics as met
```

[ ]:

```
# Keicimen = 0
# Besni  = 1

# ----BEFORE CLEANING----
names = ['area', 'major', 'minor', 'ecc', 'convex', 'extent', 'perimeter',␣
 ↪'class']
dataset = read_csv('Raisin_Dataset_CSV.csv', header=0, names=names)
# summarize the dataset
print(dataset.describe())

# replace class names with 0 or 1 values for classification
dataset = dataset.replace('Kecimen', 0)
dataset = dataset.replace('Besni', 1)
print(dataset.sample(20))
```

```
                area       major       minor         ecc         convex  \
count     900.000000  900.000000  900.000000  900.000000     900.000000
mean    87804.127778  430.929950  254.488133    0.781542   91186.090000
std     39002.111390  116.035121   49.988902    0.090318   40769.290132
min     25387.000000  225.629541  143.710872    0.348730   26139.000000
25%     59348.000000  345.442898  219.111126    0.741766   61513.250000
50%     78902.000000  407.803951  247.848409    0.798846   81651.000000
75%    105028.250000  494.187014  279.888575    0.842571  108375.750000
max    235047.000000  997.291941  492.275279    0.962124  278217.000000

           extent    perimeter       class
count  900.000000   900.000000  900.000000
mean     0.699508  1165.906636    0.500000
std      0.053468   273.764315    0.500278
min      0.379856   619.074000    0.000000
25%      0.670869   966.410750    0.000000
50%      0.707367  1119.509000    0.500000
75%      0.734991  1308.389750    1.000000
max      0.835455  2697.753000    1.000000
       area       major       minor       ecc  convex    extent  perimeter  \
730   89431  464.833932  247.352044  0.846663   92412  0.624797   1198.853
414   37302  284.330112  170.533002  0.800172   39280  0.721174    761.131
798   71054  364.751136  249.723369  0.728881   72956  0.727848   1027.206
771  175946  604.530068  372.849843  0.787151  180289  0.775838   1622.809
650  194864  657.867843  378.058461  0.818384  197430  0.786427   1700.937
881  150420  607.709674  316.809724  0.853363  153905  0.642409   1557.266
196   69312  373.803105  237.194577  0.772887   70719  0.706156   1001.236
430   57838  350.285459  214.802450  0.789911   59449  0.674040    939.149
79    47851  323.152519  191.610962  0.805245   49187  0.773237    860.402
```

```
369   56477  334.588138  219.185620  0.755550   58408  0.685667    941.429
518   64303  442.745314  187.029019  0.906395   67199  0.686235   1081.680
36    55827  305.298843  234.661225  0.639696   57724  0.703288    926.095
532   82161  441.795552  246.810056  0.829402   86909  0.630243   1222.158
421   90191  431.366966  272.751395  0.774727   93719  0.625024   1179.374
742  107082  536.851569  258.717160  0.876218  112201  0.729521   1354.715
252   38545  283.987539  175.420515  0.786410   39765  0.760436    770.590
129   88197  473.286690  238.738453  0.863455   90297  0.658457   1193.280
830  125968  522.543776  308.047596  0.807757  129444  0.739579   1414.078
172   58741  345.485448  222.381845  0.765295   60701  0.714819    948.233
195   87302  392.910117  284.179068  0.690568   89605  0.738246   1122.160

     class
730      1
414      0
798      1
771      1
650      1
881      1
196      0
430      0
79       0
369      0
518      1
36       0
532      1
421      0
742      1
252      0
129      0
830      1
172      0
195      0
```

[ ]:

```
# count the missing values
print(dataset.isnull().sum())
```

```
area         0
major        0
minor        0
ecc          0
convex       0
extent       0
perimeter    0
class        0
dtype: int64
```

[ ]:

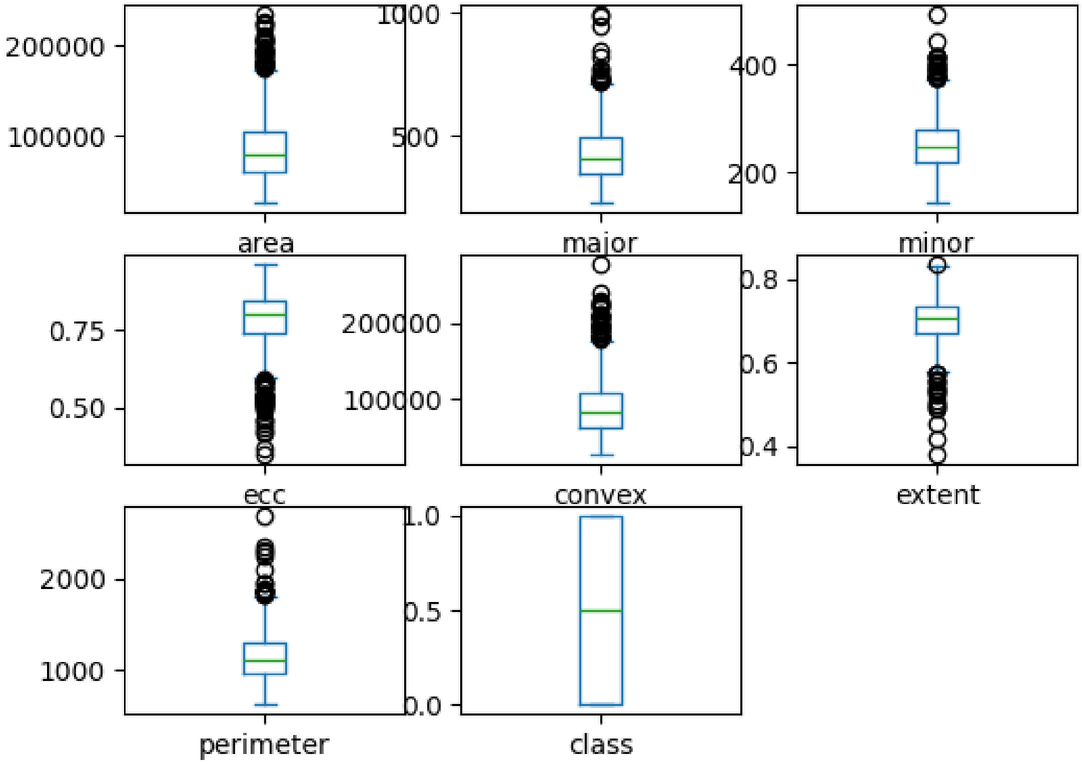
```
# locate the column with the 99999 values
num_missing = (dataset == 99999).sum()
print("Total 99999:")
print(num_missing)
# find and print the indices
for col in dataset:
    if num_missing[col] != 0:
        indices = dataset[dataset[col] == 99999].index
        print(col + ":" + str(list(indices)))

# locate the column with the 0 values
num_missing = (dataset == 0).sum()
print("Total invalid 0:")
print(num_missing)
# find and print the indices
# we can ignore the 0 values in class because 0 is a valid value
for col in dataset:
    if (col != "class"):
        if num_missing[col] != 0:
            indices = dataset[dataset[col] == 0].index
            print(col + ":" + str(list(indices)))
```

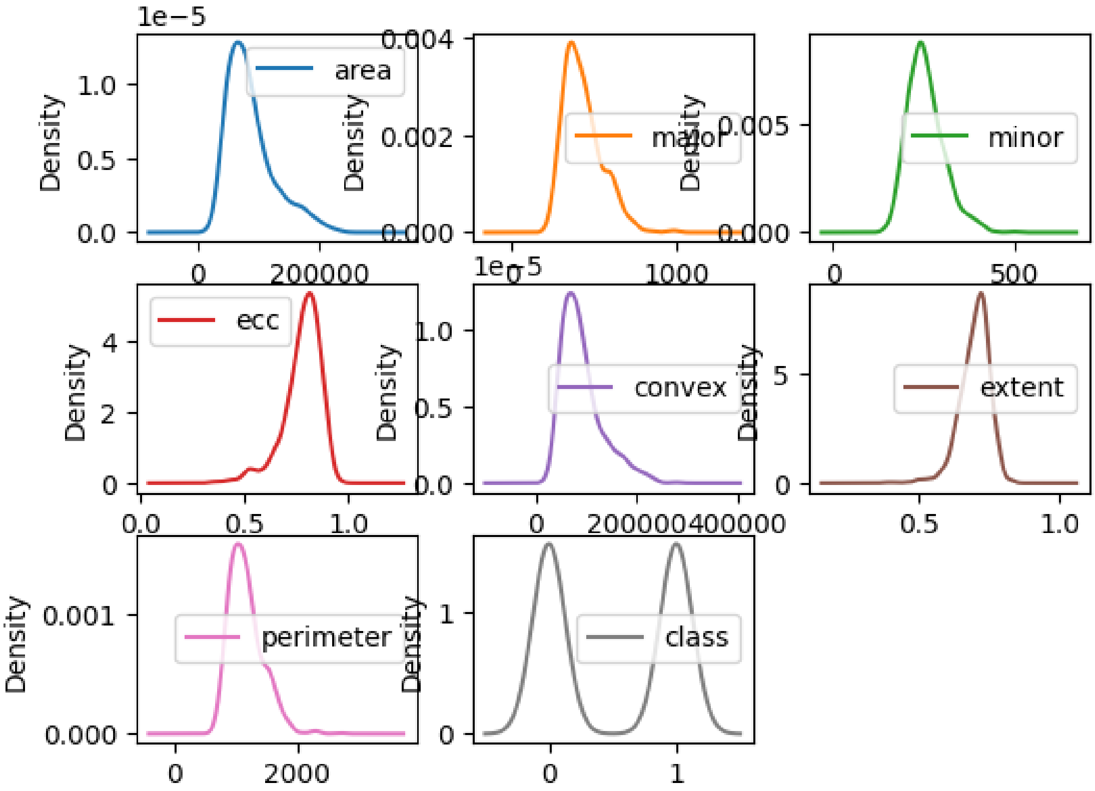
```
Total 99999:
area         0
major        0
minor        0
ecc          0
convex       0
extent       0
perimeter    0
class        0
dtype: int64
Total invalid 0:
area           0
major          0
minor          0
ecc            0
convex         0
extent         0
perimeter      0
class        450
dtype: int64
```

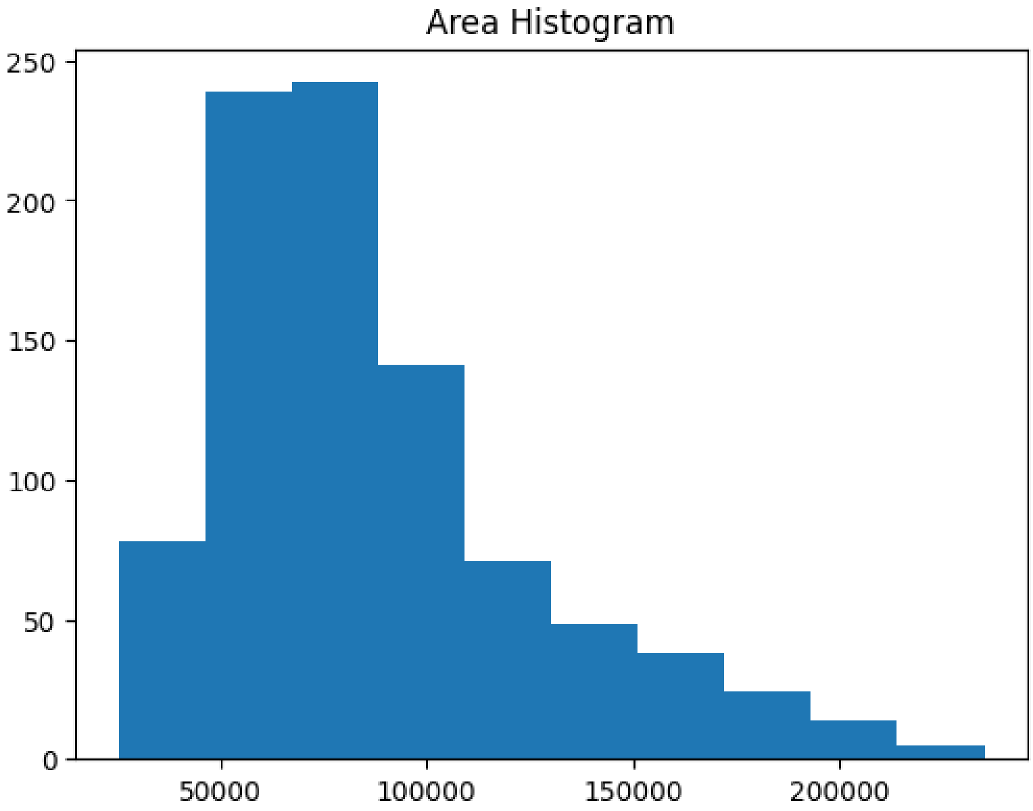
[ ]:

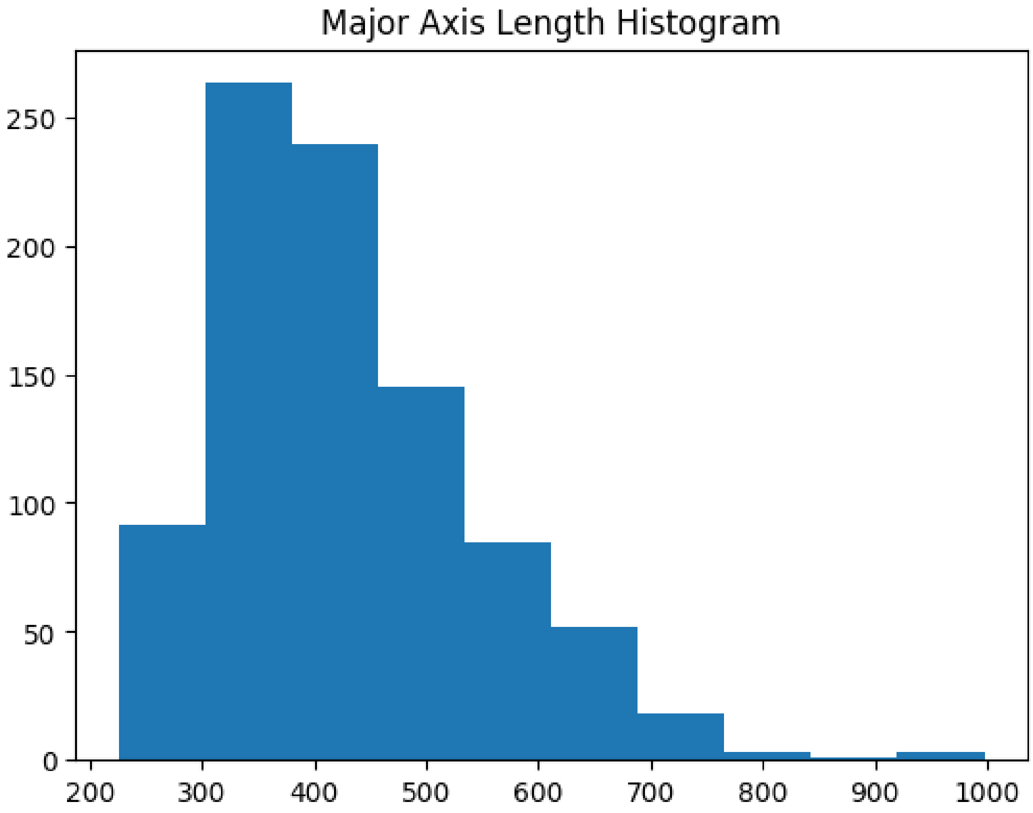
```
# ----DATA PLOTS----

# Box Plot
dataset.plot(kind='box', subplots=True, layout=(3,3), sharex=False,␣
 ↪sharey=False)
```

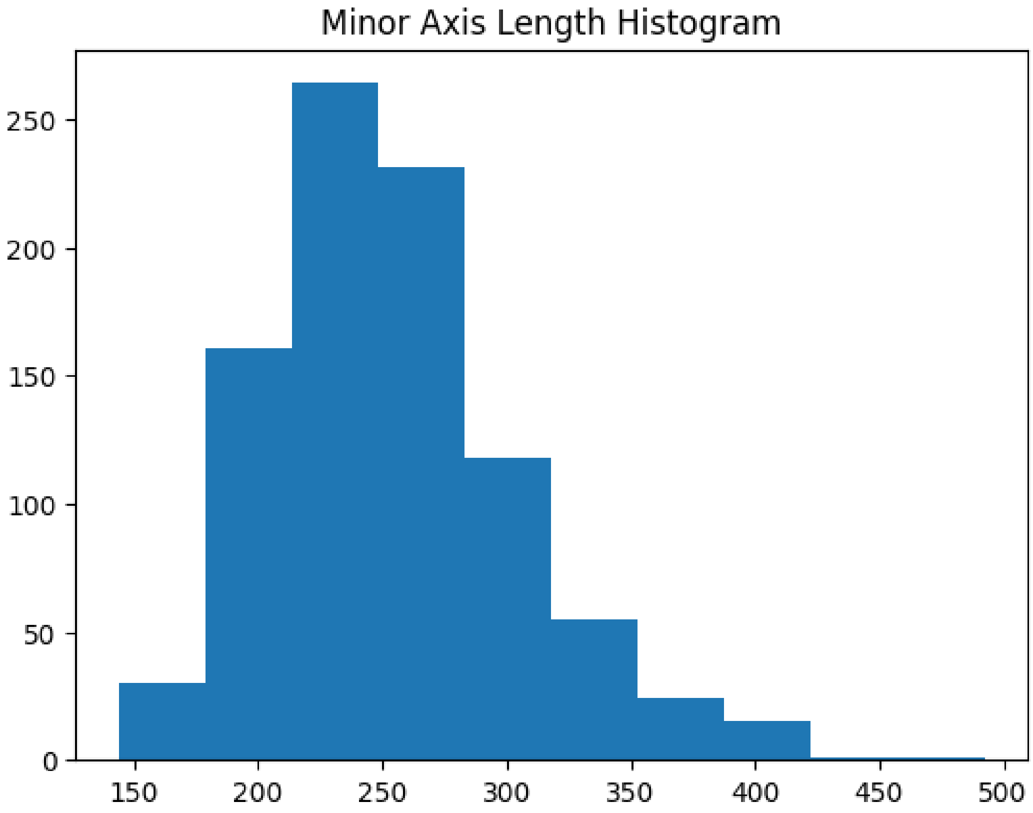
```python
plt.show()

# Density Curve
dataset.plot(kind='density', subplots=True, layout=(3, 3), sharex=False)
plt.title("Data Density Before Cleaning")
plt.show()


# Histograms
# Area Histogram
x = dataset['area']
plt.hist(x)
plt.title("Area Histogram")
plt.show()
# Major Axis Length Histogram
x = dataset['major']
plt.hist(x)
plt.title("Major Axis Length Histogram")
plt.show()
# Minor Axis Length Histogram
x = dataset['minor']
plt.hist(x)
plt.title("Minor Axis Length Histogram")
plt.show()
# Eccentricity Histogram
x = dataset['ecc']
plt.hist(x)
plt.title("Eccentricity Histogram")
plt.show()
# ConvexArea Histogram
x = dataset['convex']
plt.hist(x)
plt.title("Convex Area Histogram")
plt.show()
# Extent Histogram
x = dataset['extent']
plt.hist(x)
plt.title("Extent Histogram")
plt.show()
# Perimeter Histogram
x = dataset['perimeter']
plt.hist(x)
plt.title("Perimeter Histogram")
plt.show()
```

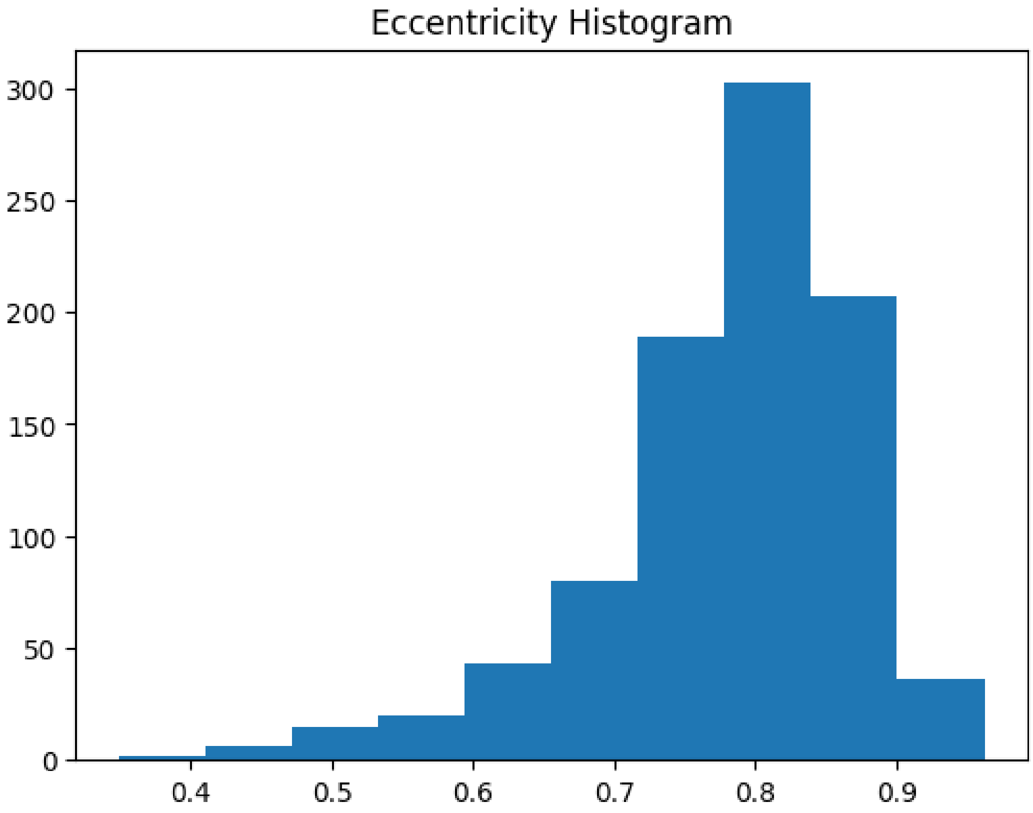

200000
100000
area
1000
500
major
400
200
minor
0.75
0.50
ecc
200000
100000
convex
0.8
0.6
0.4
extent
2000
1000
perimeter
1.0
0.5
0.0
class

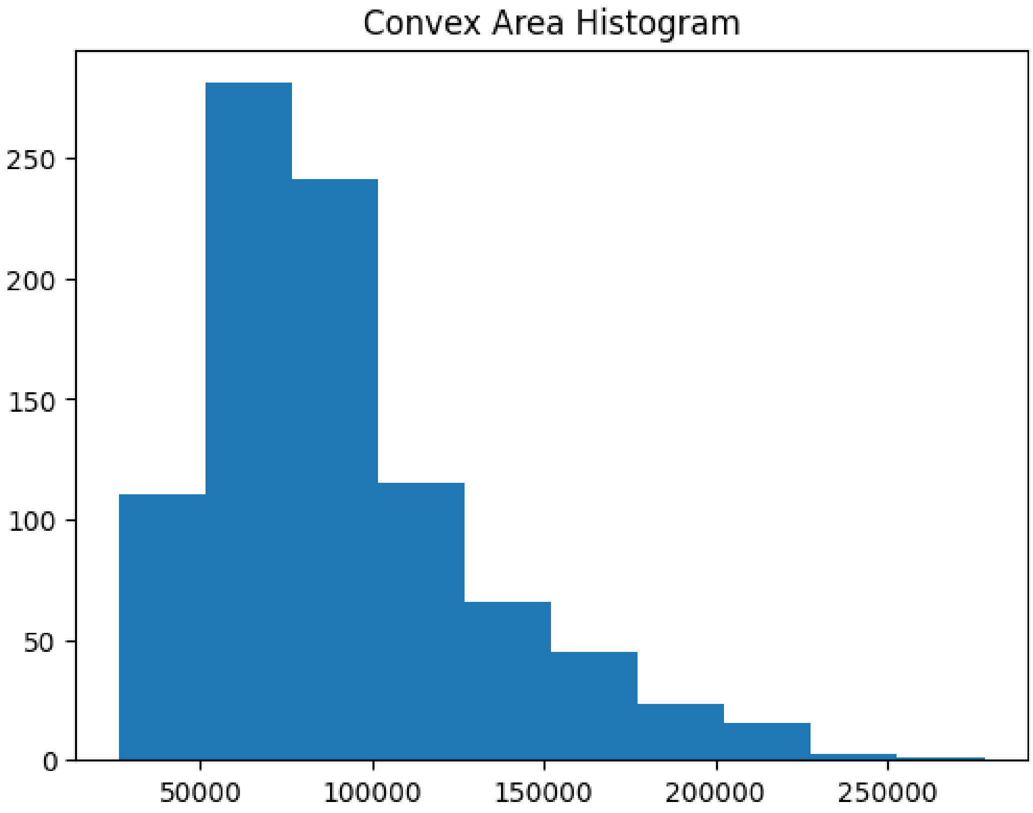

1e−5
Density
1.0
0.5
0.0
area
0
200000
Density
0.004
0.002
0.000
major
0
1000
Density
0.005
0.000
minor
0
500
Density
4
2
0
ecc
0.0
0.5
1.0
1e−5
Density
1.0
0.5
0.0
convex
0
200000
400000
Density
5
0
extent
0.5
1.0
Density
0.001
0.000
perimeter
0
2000
Density
1
0
class
0
1

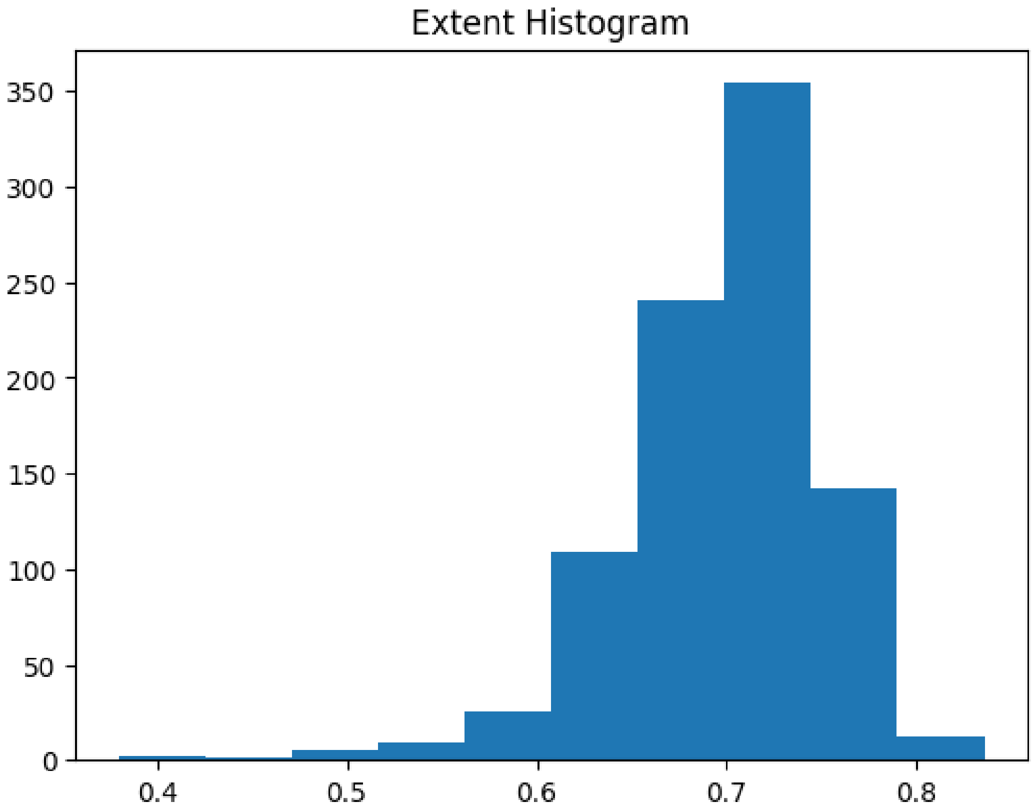

Area Histogram
250
200
150
100
50
0
50000
100000
150000
200000

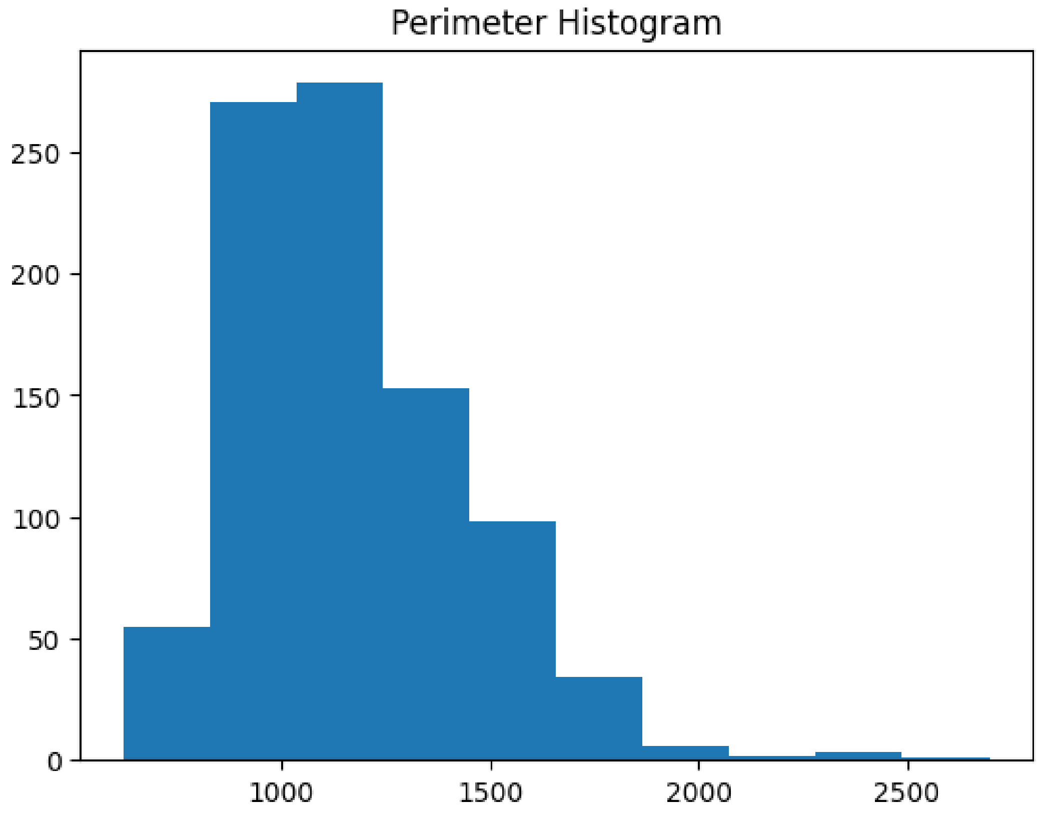

Major Axis Length Histogram
250
200
150
100
50
0
200
300
400
500
600
700
800
900
1000


Minor Axis Length Histogram
0
50
100
150
200
250
150
200
250
300
350
400
450
500


Eccentricity Histogram
300
250
200
150
100
50
0
0.4
0.5
0.6
0.7
0.8
0.9

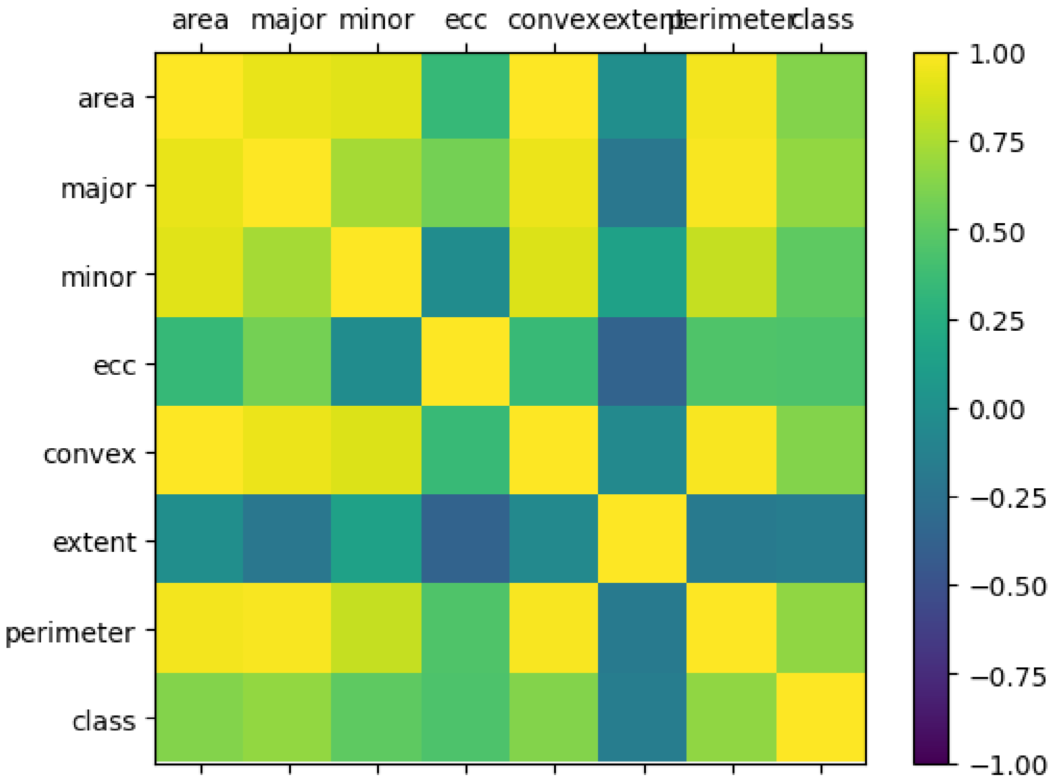

Convex Area Histogram
0
50
100
150
200
250
50000
100000
150000
200000
250000


Extent Histogram
350
300
250
200
150
100
50
0
0.4
0.5
0.6
0.7
0.8



[ ]:

```
# Correlation Matrix
correlations = dataset.corr()
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(correlations, vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = np.arange(0,8,1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
ax.set_xticklabels(names)
ax.set_yticklabels(names)
plt.show()
```



[ ]:

```
# ----Clean the Data----
threshold_z = 3
outlier_indices = []
for col in dataset:
    if col != "class":
        z = np.abs(stats.zscore(dataset[col]))
        outlier_indices_col = np.where(z > threshold_z)[0]
        outlier_indices.extend(outlier_indices_col)
print(outlier_indices)
dataset = dataset.drop(outlier_indices)
print(dataset.describe())
```

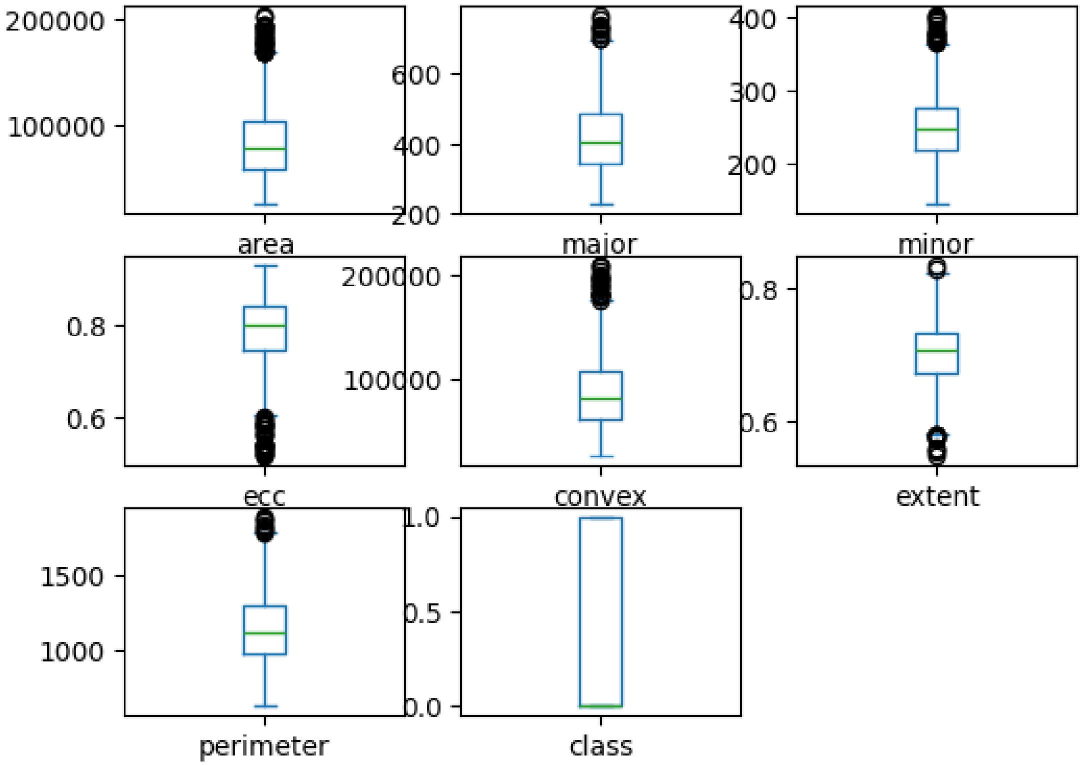
```
[487, 507, 552, 580, 581, 668, 708, 804, 812, 866, 880, 85, 487, 506, 694, 836,
490, 552, 581, 616, 769, 804, 880, 13, 23, 34, 43, 106, 156, 192, 197, 235, 251,
253, 316, 356, 483, 880, 85, 487, 506, 694, 708, 804, 812, 836, 866, 880, 85,
290, 459, 487, 498, 506, 536, 579, 682, 694, 802, 829, 836, 85, 290, 487, 506,
694, 836]
               area       major       minor         ecc        convex  \
count    860.000000  860.000000  860.000000  860.000000    860.000000
mean   85689.111628  425.399183  251.839739    0.785030  88762.722093
std    35714.789417  105.363100   46.579123    0.079252  36697.834521
min    25387.000000  225.629541  143.710872    0.517135  26139.000000
```

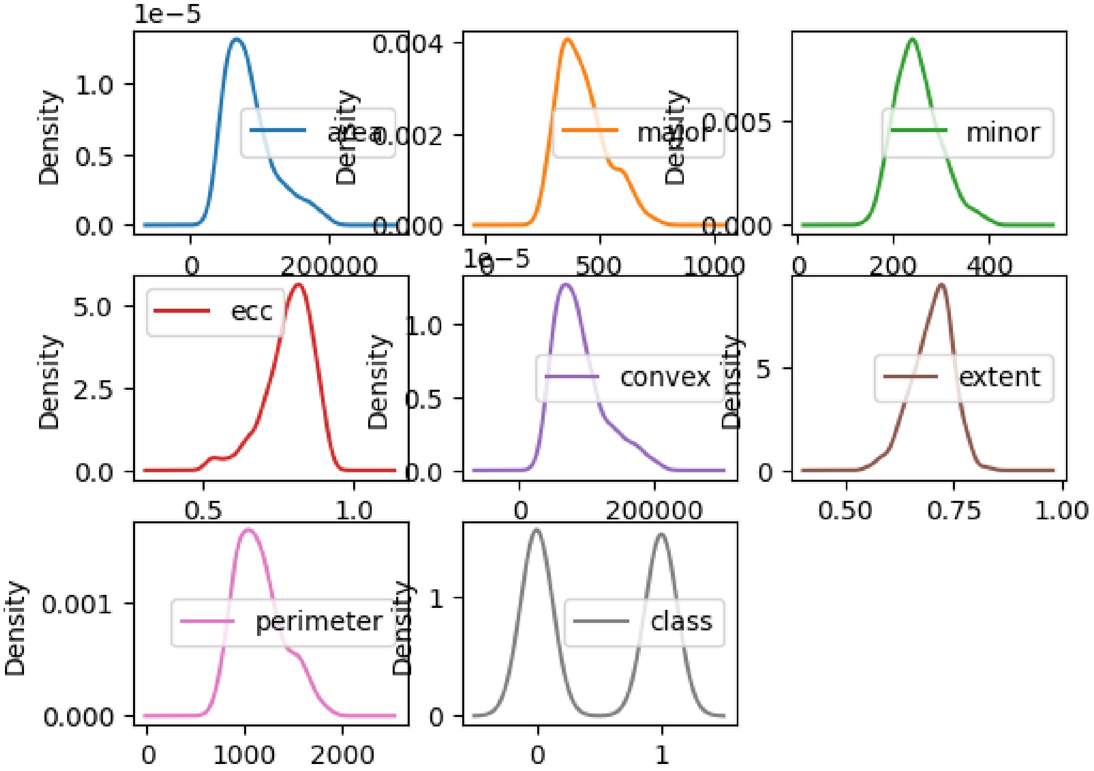
```
25%     59230.500000  345.829099  218.450742    0.744472   61489.250000
50%     78122.500000  405.796609  246.643156    0.798846   80835.500000
75%    103121.000000  486.582391  277.440665    0.840973  107340.500000
max    204226.000000  769.425149  403.719328    0.928094  209580.000000

           extent    perimeter       class
count  860.000000   860.000000  860.000000
mean     0.701263  1150.554856    0.494186
std      0.046813   245.921977    0.500257
min      0.547433   619.074000    0.000000
25%      0.671999   966.013750    0.000000
50%      0.706631  1113.375500    0.000000
75%      0.733665  1295.296750    1.000000
max      0.835455  1893.414000    1.000000
```

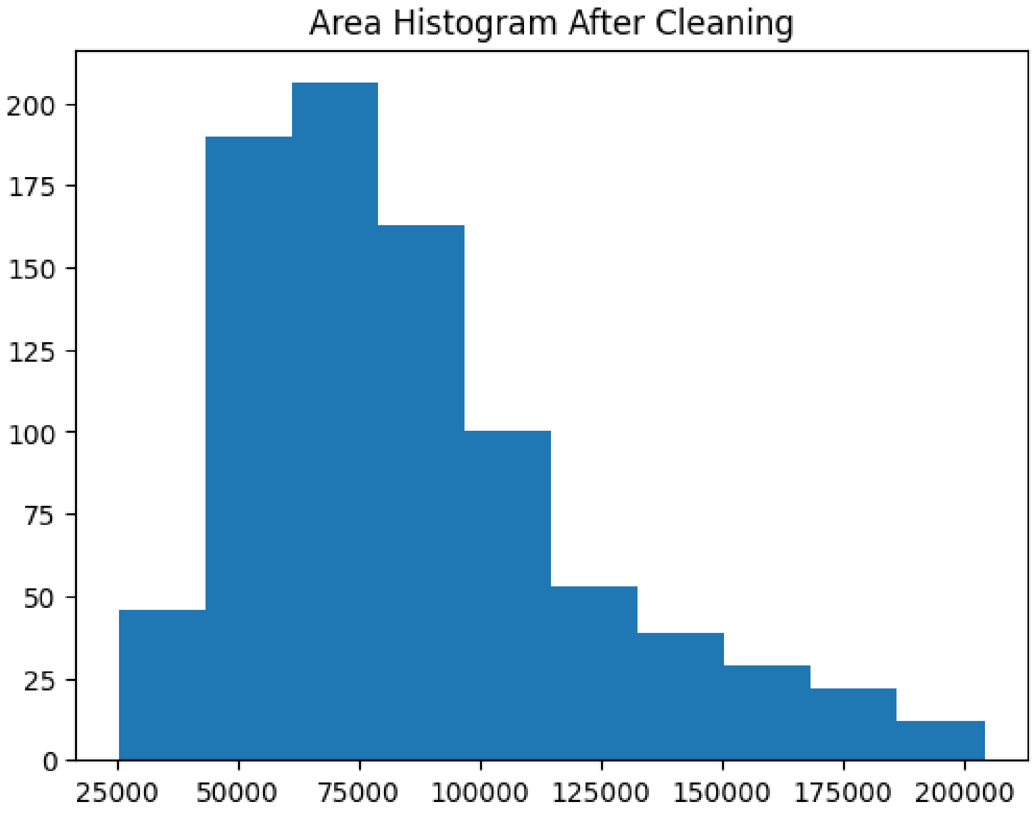
[ ]:

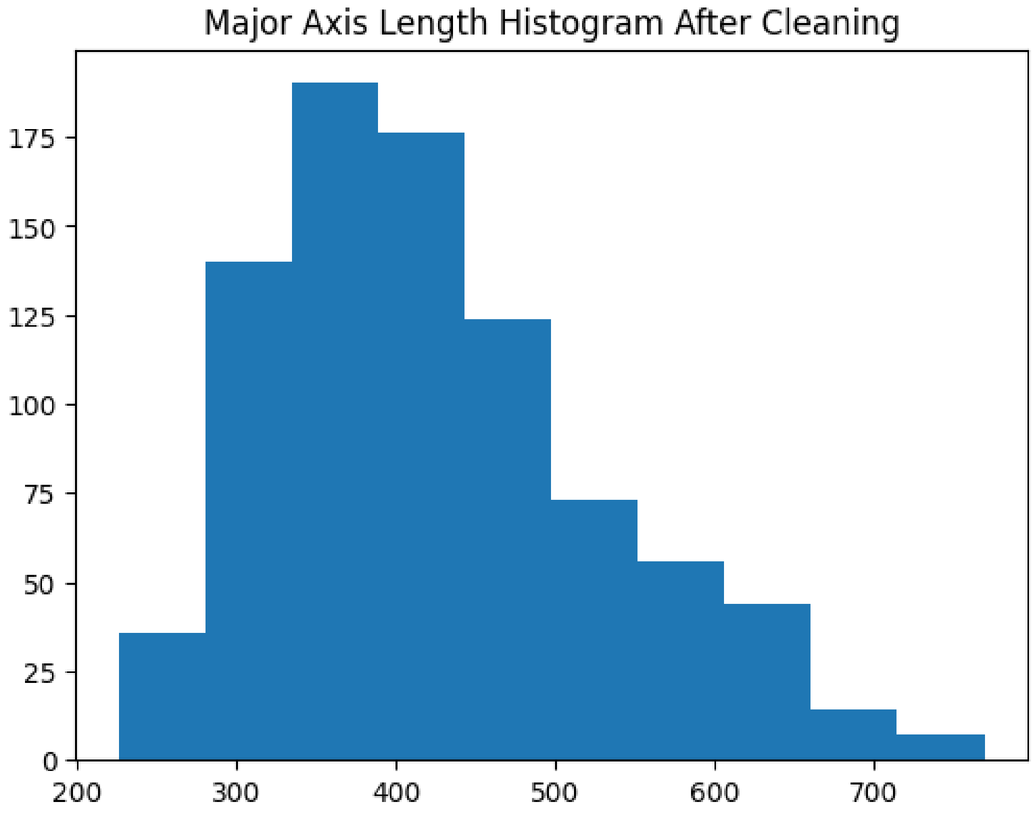
```
# ----DATA PLOTS AFTER CLEANING----

# Box Plot
dataset.plot(kind='box', subplots=True, layout=(3,3), sharex=False,␣
 ↪sharey=False)
plt.show()



# Density Plot
dataset.plot(kind='density', subplots=True, layout=(3, 3), sharex=False)
plt.title("Data Density After Cleaning")
plt.show()


# Histograms
# Area Histogram
x = dataset['area']
plt.hist(x)
plt.title("Area Histogram After Cleaning")
plt.show()
# Major Axis Length Histogram
x = dataset['major']
plt.hist(x)
plt.title("Major Axis Length Histogram After Cleaning")
plt.show()
# Minor Axis Length Histogram
x = dataset['minor']
plt.hist(x)
plt.title("Minor Axis Length Histogram After Cleaning")
plt.show()
# Eccentricity Histogram
```

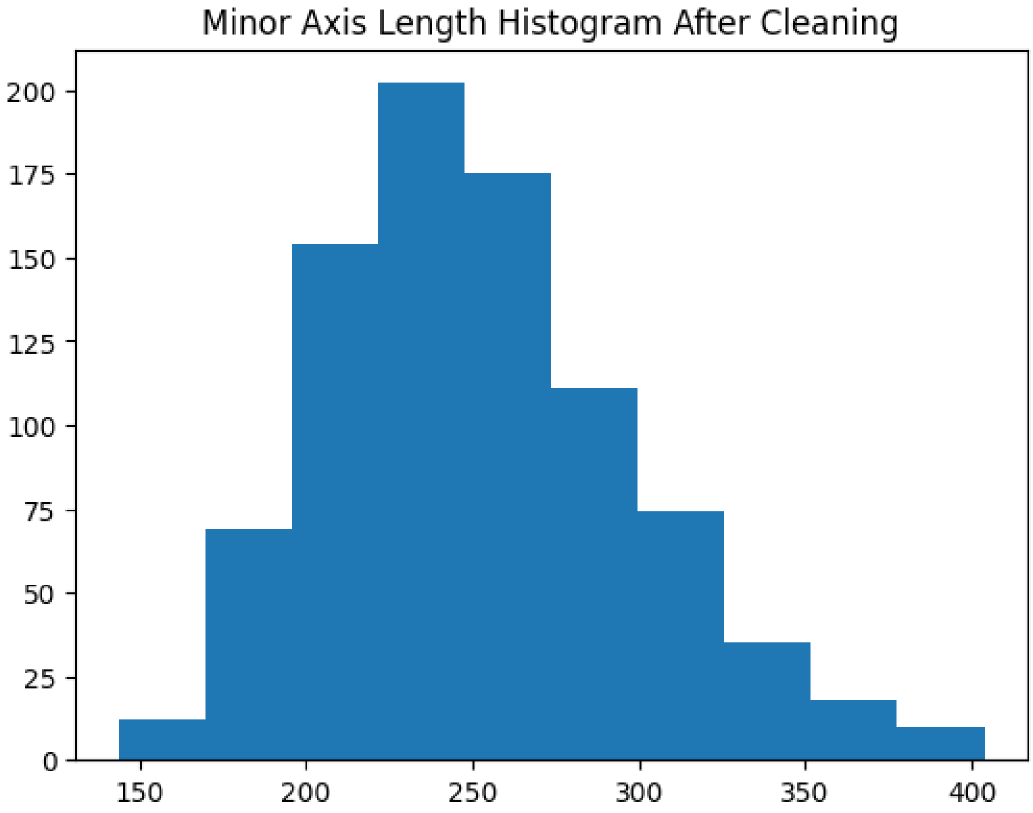
```
x = dataset['ecc']
plt.hist(x)
plt.title("Eccentricity Histogram After Cleaning")
plt.show()
# ConvexArea Histogram
x = dataset['convex']
plt.hist(x)
plt.title("Convex Area Histogram After Cleaning")
plt.show()
# Extent Histogram
x = dataset['extent']
plt.hist(x)
plt.title("Extent Histogram After Cleaning")
plt.show()
# Perimeter Histogram
x = dataset['perimeter']
plt.hist(x)
plt.title("Perimeter Histogram After Cleaning")
plt.show()
```

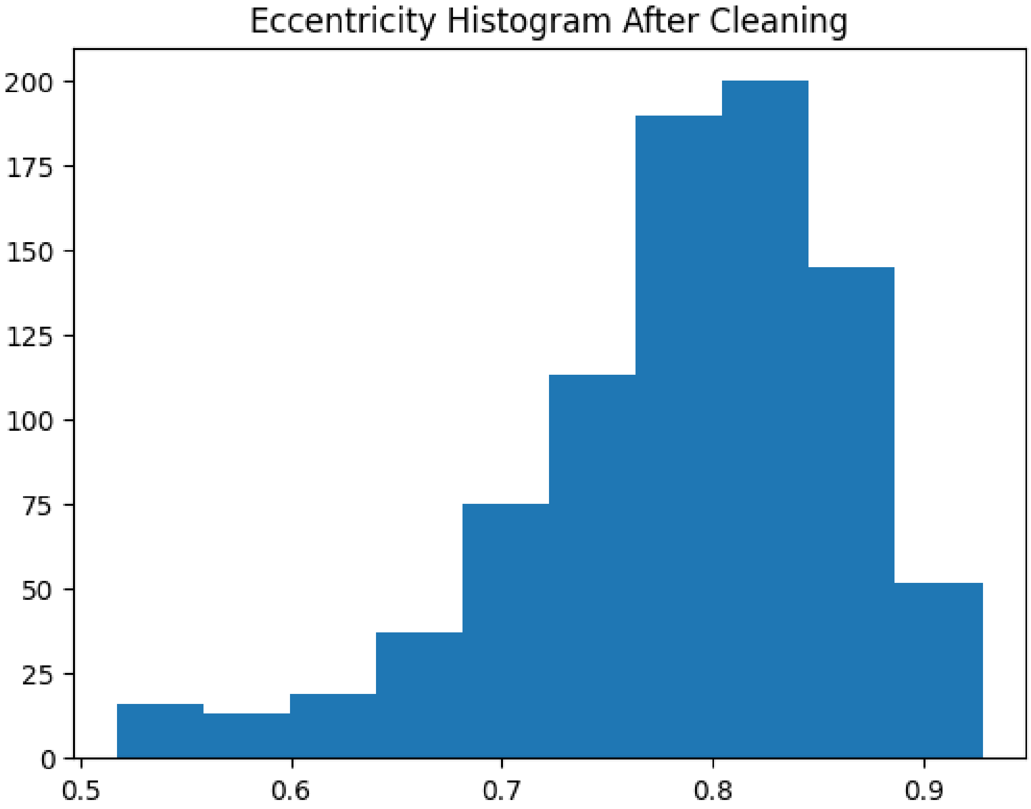


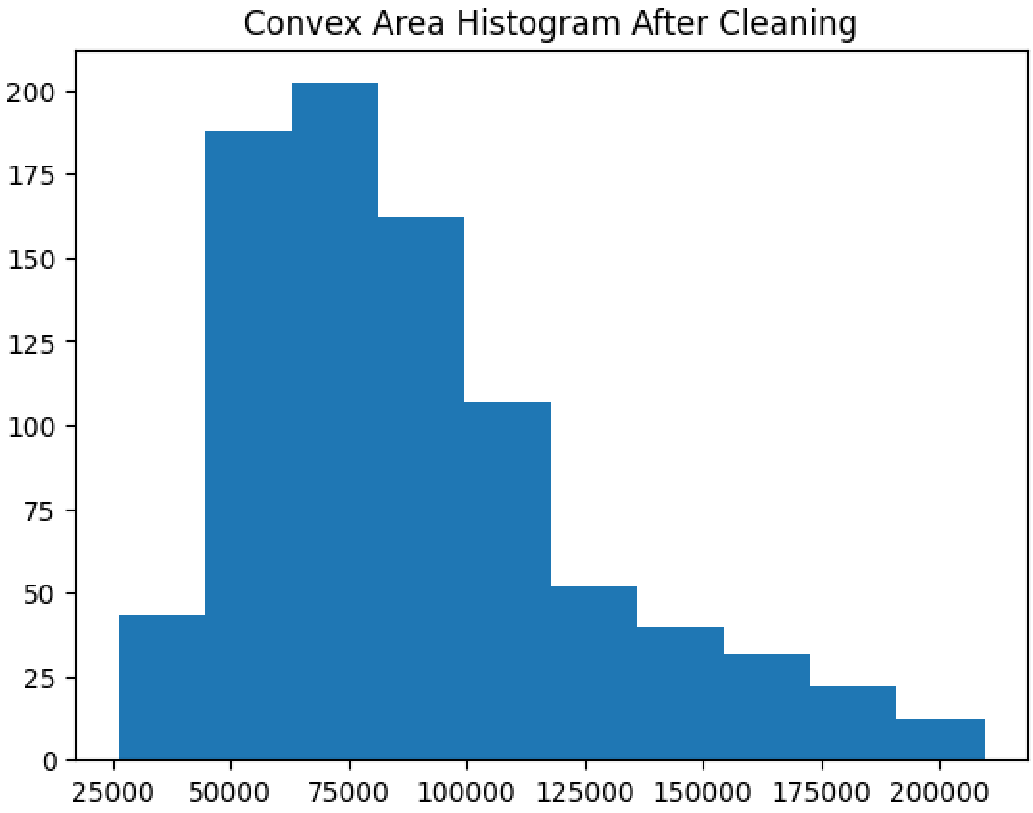

1e−5
Density
1.0
0.5
0.0
0
200000
area
0.004
0.002
0.000
Density
0
500
1000
major
0.005
0.000
Density
0
200
400
minor
5.0
2.5
0.0
Density
0.5
1.0
ecc
1e−5
1.0
0.5
0.0
Density
0
200000
convex
5
0
Density
0.50
0.75
1.00
extent
0.001
0.000
Density
0
1000
2000
perimeter
1
0
Density
0
1
class

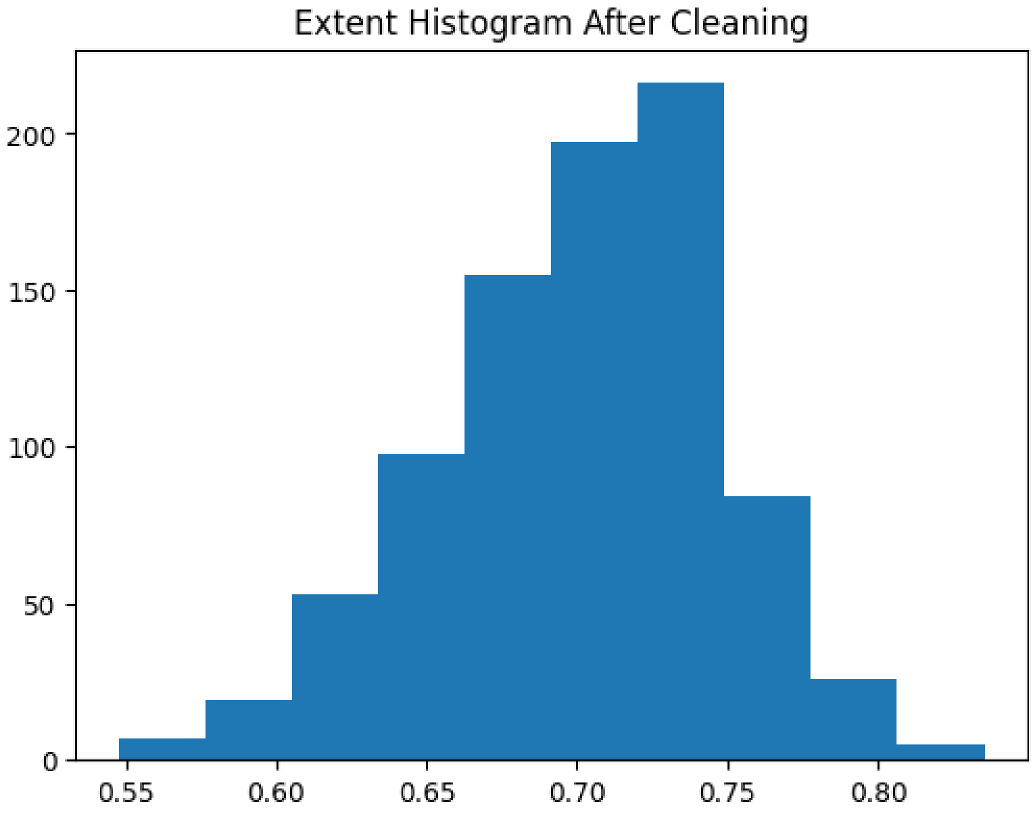

Area Histogram After Cleaning
200
175
150
125
100
75
50
25
0
25000
50000
75000
100000
125000
150000
175000
200000

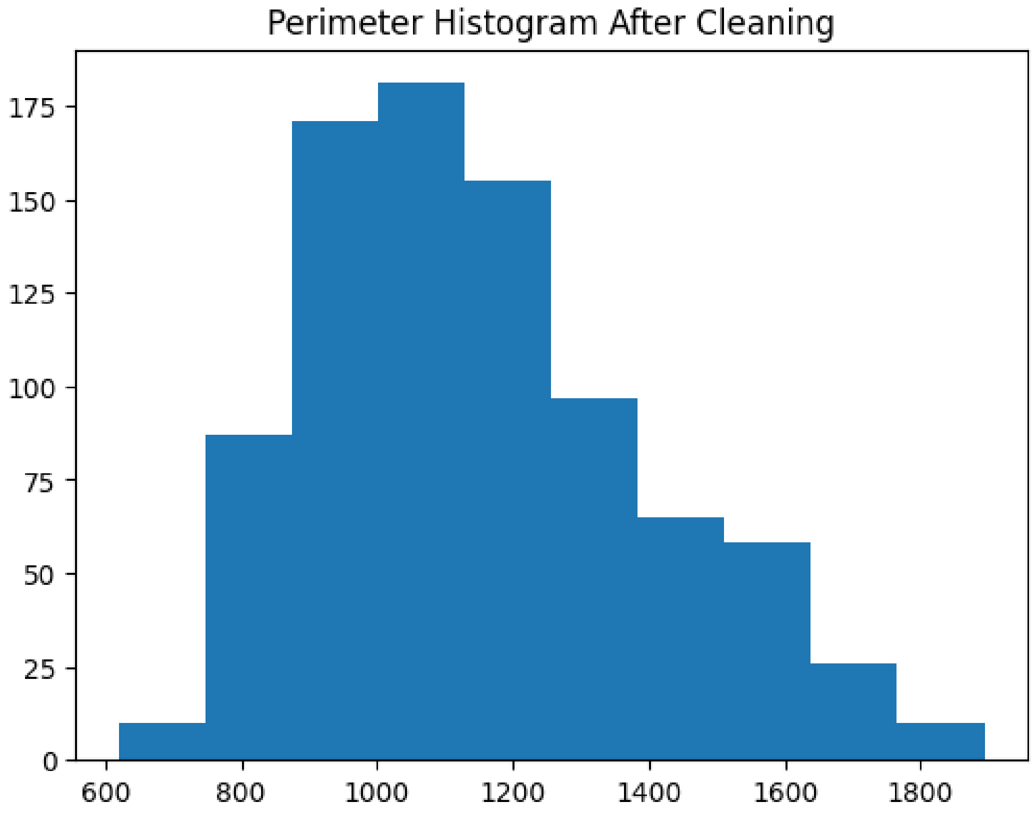

Major Axis Length Histogram After Cleaning
0
25
50
75
100
125
150
175
200
300
400
500
600
700


Minor Axis Length Histogram After Cleaning
200
175
150
125
100
75
50
25
0
150
200
250
300
350
400

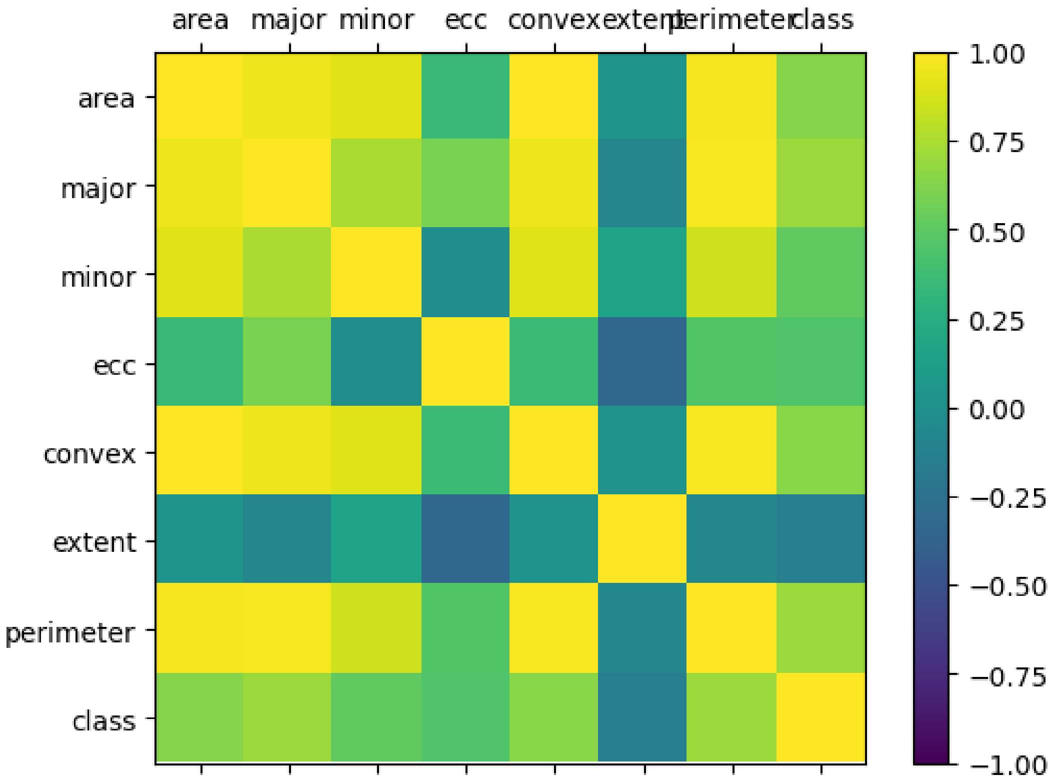

Eccentricity Histogram After Cleaning
200
175
150
125
100
75
50
25
0
0.5
0.6
0.7
0.8
0.9


Convex Area Histogram After Cleaning
200
175
150
125
100
75
50
25
0
25000
50000
75000
100000
125000
150000
175000
200000


Extent Histogram After Cleaning
0
50
100
150
200
0.55
0.60
0.65
0.70
0.75
0.80



```
[ ]: # Correlation Matrix
correlations = dataset.corr()
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(correlations, vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = np.arange(0,8,1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
ax.set_xticklabels(names)
ax.set_yticklabels(names)
plt.show()
```


area
major
minor
ecc
convex
extent
perimeter
class
area
major
minor
ecc
convex
extent
perimeter
class
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00