CSC 767 Neural Networks and Deep Learning
Homework # 1
Part 2

Team Members:

Loading data and summarize data

```
In [12]: # Listing 6

         # Load libraries
         from matplotlib import pyplot
         from pandas import read_csv
         from pandas import set_option
         from pandas.plotting import scatter_matrix
         import numpy
         from numpy import set_printoptions
         from sklearn.preprocessing import MinMaxScaler
         from sklearn.preprocessing import StandardScaler
         from sklearn.preprocessing import Normalizer
         from sklearn.preprocessing import Binarizer
```

```
In [14]: # Load dataset
         url = 'winequality-white.csv'
         names = ['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides',
                  'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol',
                  'quality']
         dataset = read_csv(url, sep=';')


         # Summarize Data

         # Descriptive statistics
         # shape
         print(dataset.shape)

         # List all data types used by the DataFrame to characterize each attribute using dtypes property.

         set_option('display.max_rows', 500)
         print(dataset.dtypes)
         # View first 20 rows
         set_option('display.width', 100)
         print(dataset.head(20))
         # descriptions, change precision to 3 places
         set_option('precision', 3)
```

(4898, 12)
fixed acidity   float64
volatile acidity   float64

citric acid	float64
residual sugar	float64
chlorides	float64
free sulfur dioxide	float64
total sulfur dioxide	float64
density	float64
pH	float64
sulphates	float64
alcohol	float64
quality	int64

dtype: object

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide \
0	7.0	0.27	0.36	20.70	0.045	45.0
1	6.3	0.30	0.34	1.60	0.049	14.0
2	8.1	0.28	0.40	6.90	0.050	30.0
3	7.2	0.23	0.32	8.50	0.058	47.0
4	7.2	0.23	0.32	8.50	0.058	47.0
5	8.1	0.28	0.40	6.90	0.050	30.0
6	6.2	0.32	0.16	7.00	0.045	30.0
7	7.0	0.27	0.36	20.70	0.045	45.0
8	6.3	0.30	0.34	1.60	0.049	14.0
9	8.1	0.22	0.43	1.50	0.044	28.0
10	8.1	0.27	0.41	1.45	0.033	11.0
11	8.6	0.23	0.40	4.20	0.035	17.0
12	7.9	0.18	0.37	1.20	0.040	16.0
13	6.6	0.16	0.40	1.50	0.044	48.0
14	8.3	0.42	0.62	19.25	0.040	41.0
15	6.6	0.17	0.38	1.50	0.032	28.0
16	6.3	0.48	0.04	1.10	0.046	30.0
17	6.2	0.66	0.48	1.20	0.029	29.0
18	7.4	0.34	0.42	1.10	0.033	17.0
19	6.5	0.31	0.14	7.50	0.044	34.0

	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	170.0	1.0010	3.00	0.45	8.8	6
1	132.0	0.9940	3.30	0.49	9.5	6
2	97.0	0.9951	3.26	0.44	10.1	6
3	186.0	0.9956	3.19	0.40	9.9	6
4	186.0	0.9956	3.19	0.40	9.9	6
5	97.0	0.9951	3.26	0.44	10.1	6
6	136.0	0.9949	3.18	0.47	9.6	6
7	170.0	1.0010	3.00	0.45	8.8	6
8	132.0	0.9940	3.30	0.49	9.5	6

9	129.0	0.9938	3.22	0.45	11.0	6
10	63.0	0.9908	2.99	0.56	12.0	5
11	109.0	0.9947	3.14	0.53	9.7	5
12	75.0	0.9920	3.18	0.63	10.8	5
13	143.0	0.9912	3.54	0.52	12.4	7
14	172.0	1.0002	2.98	0.67	9.7	5
15	112.0	0.9914	3.25	0.55	11.4	7
16	99.0	0.9928	3.24	0.36	9.6	6
17	75.0	0.9892	3.33	0.39	12.8	8
18	171.0	0.9917	3.12	0.53	11.3	6
19	133.0	0.9955	3.22	0.50	9.5	5

In [15]:
```
# The describe() function list 8 statistical properties of each attribute.
print(dataset.describe())
```

```
       fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
count       4898.000          4898.000     4898.000        4898.000   4898.000
mean           6.855             0.278        0.334           6.391      0.046
std            0.844             0.101        0.121           5.072      0.022
min            3.800             0.080        0.000           0.600      0.009
25%            6.300             0.210        0.270           1.700      0.036
50%            6.800             0.260        0.320           5.200      0.043
75%            7.300             0.320        0.390           9.900      0.050
max           14.200             1.100        1.660          65.800      0.346

       free sulfur dioxide  total sulfur dioxide  density        pH  sulphates  alcohol  \
count             4898.000              4898.000 4898.000  4898.000   4898.000 4898.000
mean                35.308               138.361    0.994     3.188      0.490   10.514
std                 17.007                42.498    0.003     0.151      0.114    1.231
min                  2.000                 9.000    0.987     2.720      0.220    8.000
25%                 23.000               108.000    0.992     3.090      0.410    9.500
50%                 34.000               134.000    0.994     3.180      0.470   10.400
75%                 46.000               167.000    0.996     3.280      0.550   11.400
max                289.000               440.000    1.039     3.820      1.080   14.200

        quality
count  4898.000
mean      5.878
std       0.886
min       3.000
25%       5.000
50%       6.000
75%       6.000
max       9.000
```

```
In [16]:
# Group class distribution
print(dataset.groupby('quality').size())
```

```
quality
3      20
4     163
5    1457
6    2198
7     880
8     175
9       5
dtype: int64
```

Listing 6a: Pairwise Pearson correlations

```
In [17]: # Correlations between attributes using Pearson's Correlation Coefficient
         # 6a. Pairwise Pearson correlations
         print(dataset.corr(method = 'pearson'))
```

Correlations between attributes using Pearson's Correlation Coefficient

Data set after applying Pairwise Pearson Correlations.

```
                      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
fixed acidity                 1.000            -0.023        0.289           0.089      0.023
volatile acidity             -0.023             1.000       -0.149           0.064      0.071
citric acid                   0.289            -0.149        1.000           0.094      0.114
residual sugar                0.089             0.064        0.094           1.000      0.089
chlorides                     0.023             0.071        0.114           0.089      1.000
free sulfur dioxide          -0.049            -0.097        0.094           0.299      0.101
total sulfur dioxide          0.091             0.089        0.121           0.401      0.199
density                       0.265             0.027        0.150           0.839      0.257
pH                           -0.426            -0.032       -0.164          -0.194     -0.090
sulphates                    -0.017            -0.036        0.062          -0.027      0.017
alcohol                      -0.121             0.068       -0.076          -0.451     -0.360
quality                      -0.114            -0.195       -0.009          -0.098     -0.210

                      free sulfur dioxide  total sulfur dioxide  density         pH  sulphates  \
fixed acidity                  -4.940e-02                 0.091    0.265 -4.259e-01     -0.017
volatile acidity               -9.701e-02                 0.089    0.027 -3.192e-02     -0.036
citric acid                     9.408e-02                 0.121    0.150 -1.637e-01      0.062
residual sugar                  2.991e-01                 0.401    0.839 -1.941e-01     -0.027
chlorides                       1.014e-01                 0.199    0.257 -9.044e-02      0.017
free sulfur dioxide             1.000e+00                 0.616    0.294 -6.178e-04      0.059
total sulfur dioxide            6.155e-01                 1.000    0.530  2.321e-03      0.135
density                         2.942e-01                 0.530    1.000 -9.359e-02      0.074
pH                             -6.178e-04                 0.002   -0.094  1.000e+00      0.156
sulphates                       5.922e-02                 0.135    0.074  1.560e-01      1.000
alcohol                        -2.501e-01                -0.449   -0.780  1.214e-01     -0.017
quality                         8.158e-03                -0.175   -0.307  9.943e-02      0.054
```

```
                      alcohol  quality
fixed acidity          -0.121   -0.114
volatile acidity        0.068   -0.195
citric acid            -0.076   -0.009
residual sugar         -0.451   -0.098
chlorides              -0.360   -0.210
free sulfur dioxide    -0.250    0.008
total sulfur dioxide   -0.449   -0.175
density                -0.780   -0.307
pH                      0.121    0.099
sulphates              -0.017    0.054
alcohol                 1.000    0.436
quality                 0.436    1.000
```

Listing 6b: Skew of Univariate Distributions

```
In [18]: # 6b.Skew of Univariate Distributions
         # Knowing an attribute has a skew may allow us to perform data preparation to correct the skew and lat
         # improve the accuracy of our models.

         print(dataset.skew())
```

Knowing an attribute has a skew may allow us to perform data preparation to correct the skew and later improve the accuracy of our models.

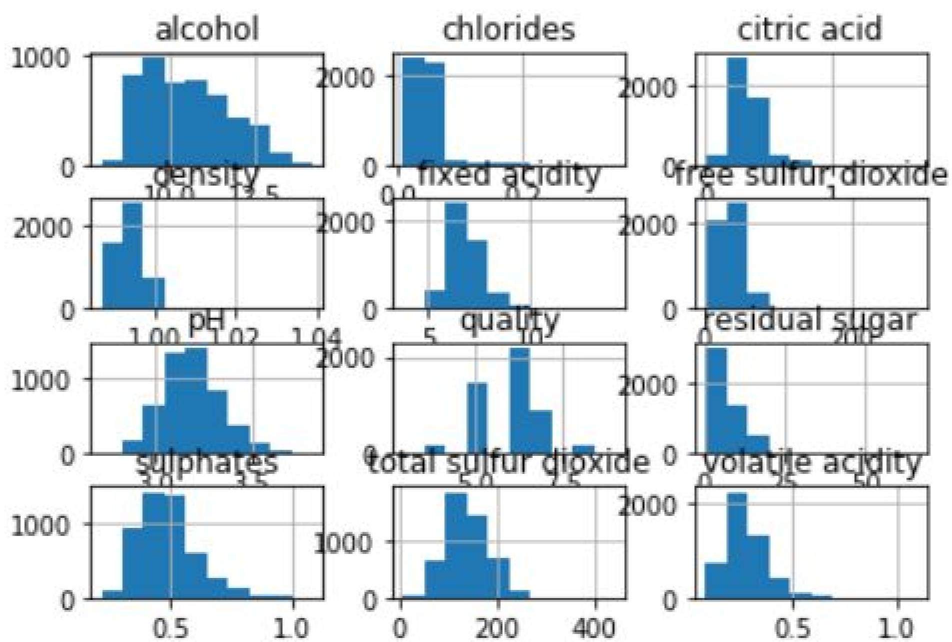
```
fixed acidity           0.648
volatile acidity        1.577
citric acid             1.282
residual sugar          1.077
chlorides               5.023
free sulfur dioxide     1.407
total sulfur dioxide    0.391
density                 0.978
pH                      0.458
sulphates               0.977
alcohol                 0.487
quality                 0.156
dtype: float64
```

Listing 6c: Univariate Density Plot

```
In [19]: # 6c. Visualization data with Univariate Plot
         # Histograms group data into bin and provide us a count of the number of observations in each bin.
         print(dataset.hist())
         pyplot.figsize = (8,8)
         pyplot.savefig('histograms.png', dpi=300)
         pyplot.show()
```

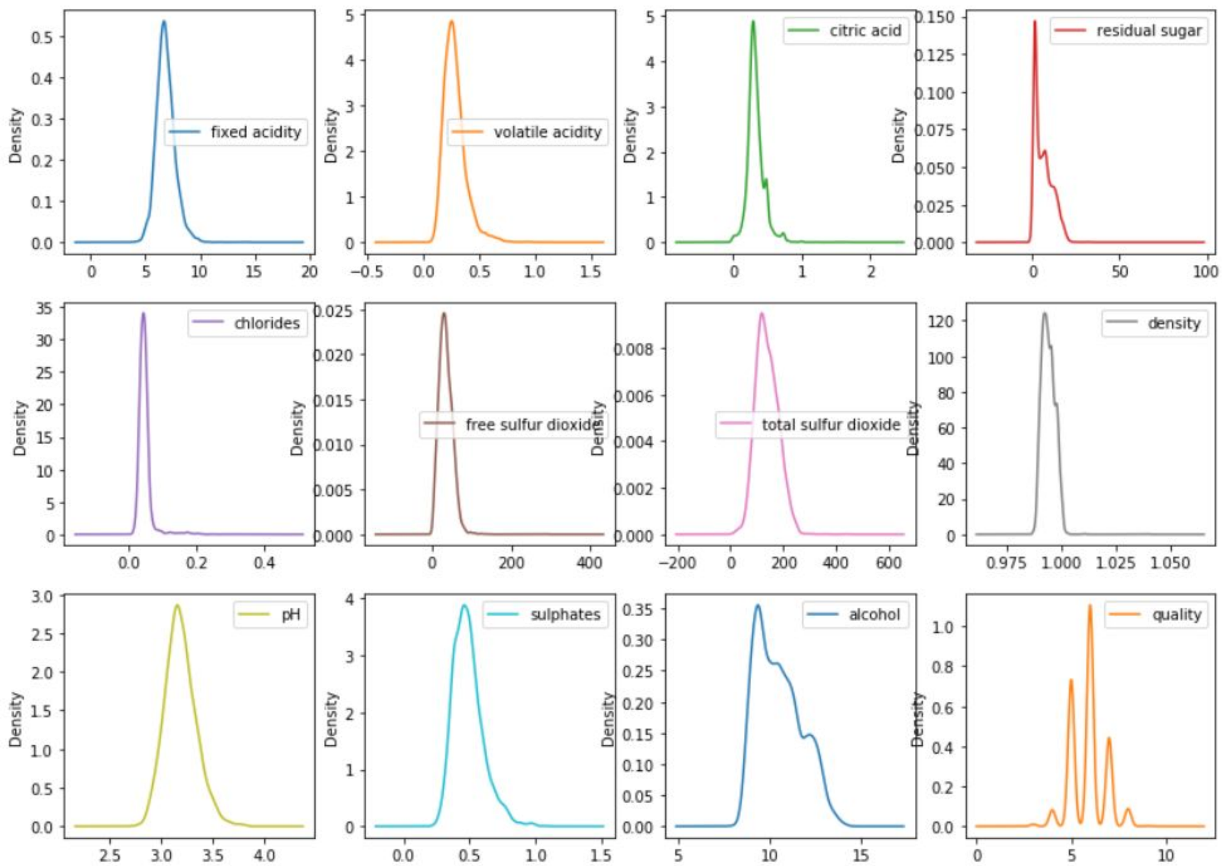
Visualization data with Univariate Plot

Histograms group data into bin and provide us a count of the number of observations in each bin.



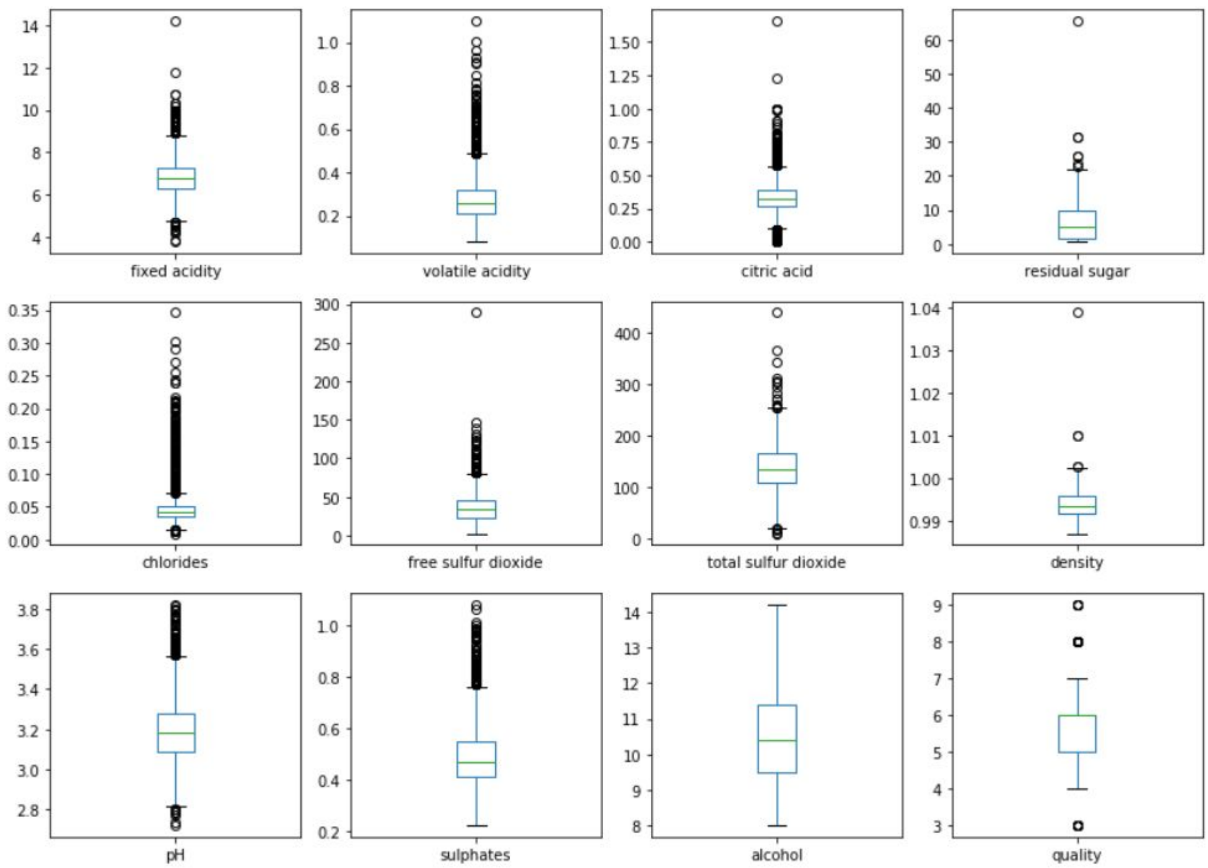
```
In [20]: # Density plots, this help us getting a quick idea of the distribution of each attribute.
         # As we can see the distribution for each attribute is clearer than the histograms

         dataset.plot(kind='density', subplots=True, layout=(4,4),sharex=False, figsize = (14,14))
         pyplot.show()
```

Density plots, this help us getting a quick idea of the distribution of each attribute.
As we can see the distribution for each attribute is clearer than the histograms



In [21]:
```
# Box and Whisker Plots, this give an idea of the spread of data and dot outside of the Whisker show
# candidate outliner values.
dataset.plot(kind='box', subplots=True, layout=(4,4),sharex=False, figsize = (14,14))
pyplot.show()
```

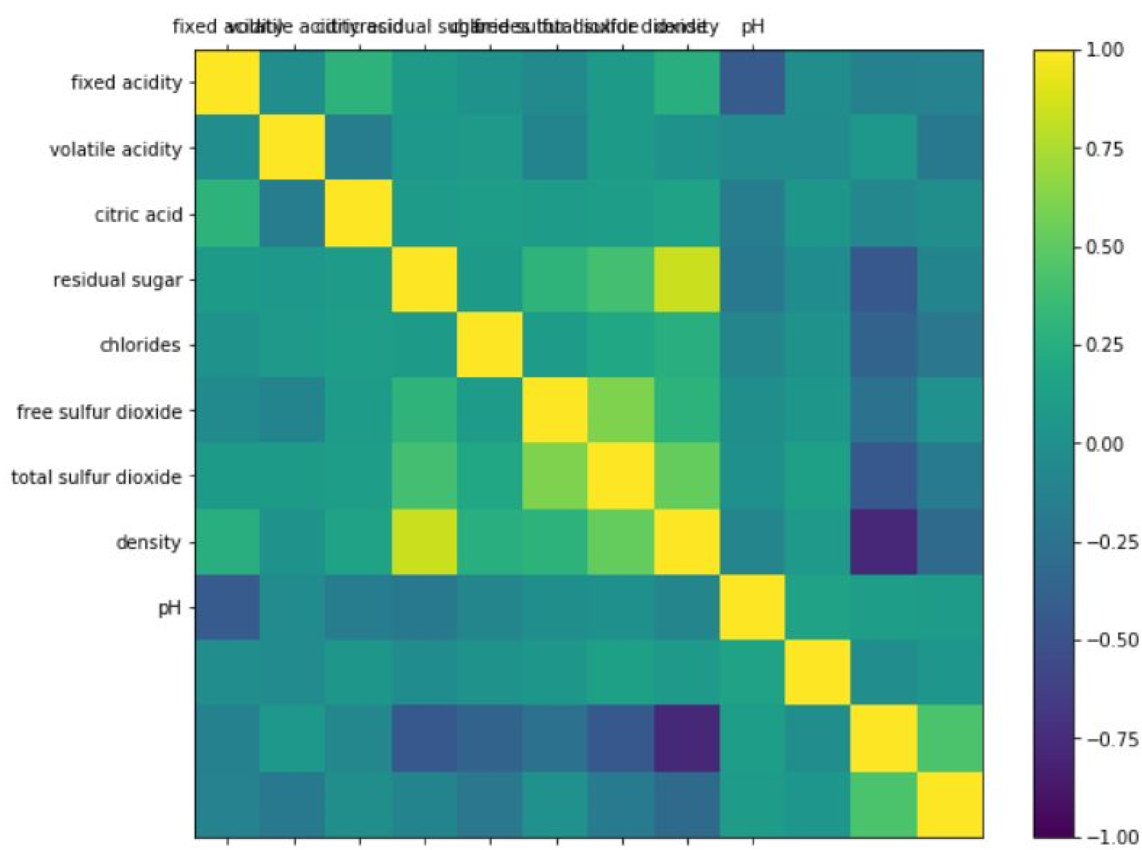
Box and Whisker Plots, this give an idea of the spread of data and dot outside of the Whisker show
candidate outlier values.



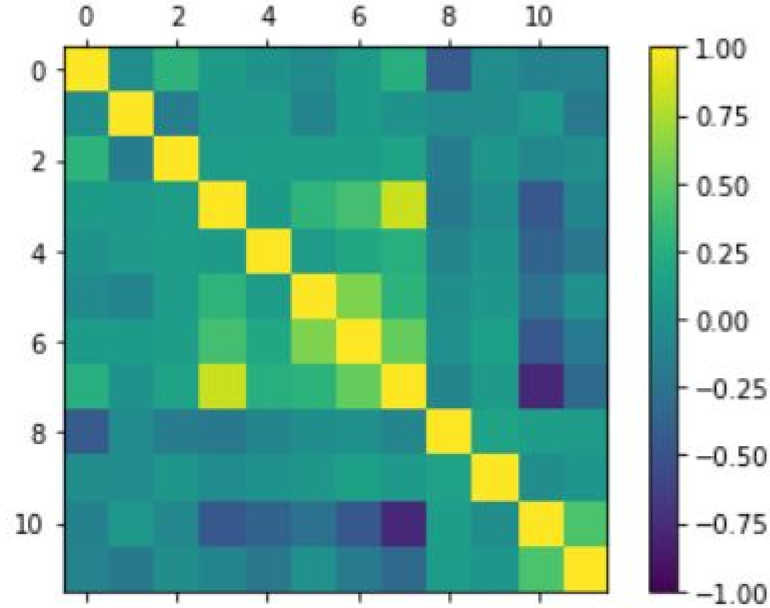
Listing 6d: Correlation Matrix Plot

In [30]:
```
# 6d. Correlation Matrix Plot
# This gives an indication of how related the changes are between two variables.
# Plot correlation matrix
fig = pyplot.figure(figsize=(10,8))
ax = fig.add_subplot(111)
cax = ax.matshow(dataset.corr(), vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = numpy.arange(0, 9, 1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
ax.set_xticklabels(names)
ax.set_yticklabels(names)
pyplot.show()
```

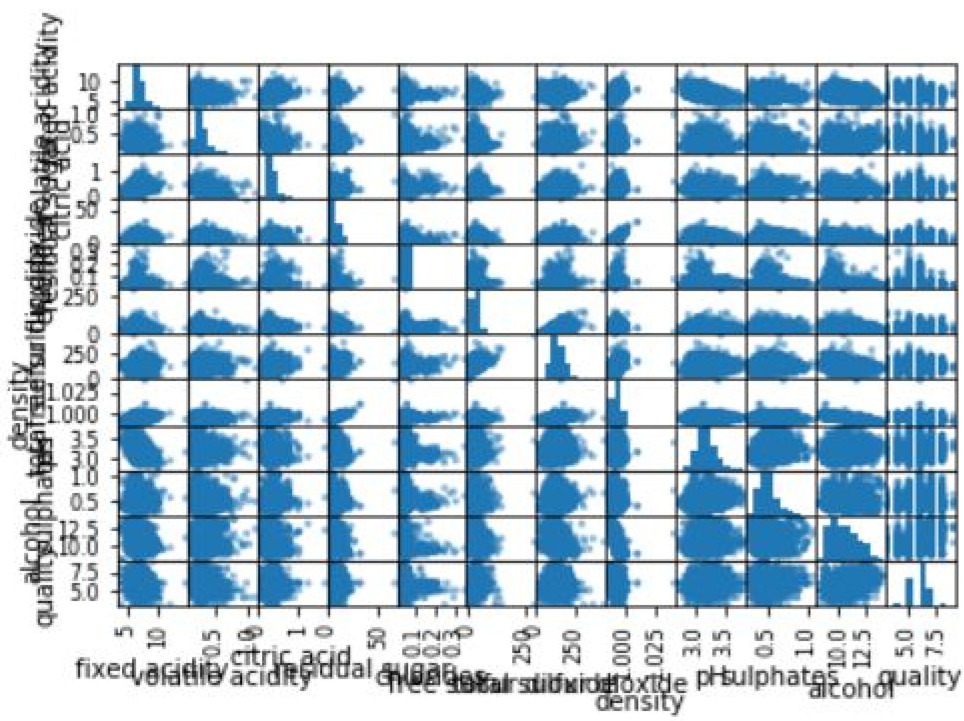
This gives an indication of how related the changes are between two variables.
Plot correlation matrix



In [26]:
```
# Correlation Matrix Generic Plot
fig = pyplot.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(dataset.corr(), vmin=-1, vmax=1)
fig.colorbar(cax)
pyplot.show()
```



In [35]:
```
# Scatterplot Maxtrix, this shows the relationship between variables
scatter_matrix(dataset)
pyplot.figure(figsize=(20,18))

pyplot.show()
```



Listing 7a: Rescaling Data

```
In [82]: # Listing 7
         # 7a. Rescaling data
         # After rescalling we can see that all of the values are in the range between 0 and 1.
         array = dataset.values
         # seperate array into input and output components
         X = array[:, 0:11]
         Y = array[:,11]
         scaler = MinMaxScaler(feature_range=(0, 1))
         rescaledX = scaler.fit_transform(X)
         set_printoptions(precision=3)
         print(rescaledX[0:5, :])
```

After rescalling we can see that all of the values are in the range between 0 and 1.

```
[[0.308 0.186 0.217 0.308 0.107 0.15  0.374 0.268 0.255 0.267 0.129]
 [0.24  0.216 0.205 0.015 0.119 0.042 0.285 0.133 0.527 0.314 0.242]
 [0.413 0.196 0.241 0.097 0.122 0.098 0.204 0.154 0.491 0.256 0.339]
 [0.327 0.147 0.193 0.121 0.145 0.157 0.411 0.164 0.427 0.209 0.306]
 [0.327 0.147 0.193 0.121 0.145 0.157 0.411 0.164 0.427 0.209 0.306]]
```

Listing 7b: Standardize Data

```
In [84]: # 7b. Standardize Data
         X = array[:, 0:11]
         Y = array[:,11]
         scaler_standard = StandardScaler().fit(X)
         rescaled_standardX = scaler_standard.transform(X)
         # summarize transformed data
         set_printoptions(precision=3)
         print(rescaled_standardX[0:5, :])
```

```
[[ 1.721e-01 -8.177e-02  2.133e-01  2.821e+00 -3.536e-02  5.699e-01
   7.446e-01  2.332e+00 -1.247e+00 -3.492e-01 -1.393e+00]
 [-6.575e-01  2.159e-01  4.800e-02 -9.448e-01  1.477e-01 -1.253e+00
  -1.497e-01 -9.154e-03  7.400e-01  1.342e-03 -8.243e-01]
 [ 1.476e+00  1.745e-02  5.438e-01  1.003e-01  1.935e-01 -3.121e-01
  -9.733e-01  3.587e-01  4.751e-01 -4.368e-01 -3.367e-01]
 [ 4.091e-01 -4.787e-01 -1.173e-01  4.158e-01  5.597e-01  6.875e-01
   1.121e+00  5.259e-01  1.148e-02 -7.873e-01 -4.992e-01]
 [ 4.091e-01 -4.787e-01 -1.173e-01  4.158e-01  5.597e-01  6.875e-01
   1.121e+00  5.259e-01  1.148e-02 -7.873e-01 -4.992e-01]]
```

Listing 7c: Normalize Data

```
In [86]: # 7c. Normalize Data

         X = array[:, 0:11]
         Y = array[:,11]
         scaler = Normalizer().fit(X)
         normalizedX = scaler.transform(X)
         # summarize transformed data
         set_printoptions(precision=3)
         print(normalizedX[0:5, :])
```

```
[[3.945e-02 1.522e-03 2.029e-03 1.166e-01 2.536e-04 2.536e-01 9.580e-01
  5.641e-03 1.691e-02 2.536e-03 4.959e-02]
 [4.727e-02 2.251e-03 2.551e-03 1.200e-02 3.676e-04 1.050e-01 9.904e-01
  7.458e-03 2.476e-02 3.676e-03 7.128e-02]
 [7.891e-02 2.728e-03 3.897e-03 6.722e-02 4.871e-04 2.923e-01 9.450e-01
  9.694e-03 3.176e-02 4.287e-03 9.840e-02]
 [3.741e-02 1.195e-03 1.663e-03 4.417e-02 3.014e-04 2.442e-01 9.665e-01
  5.173e-03 1.658e-02 2.078e-03 5.144e-02]
 [3.741e-02 1.195e-03 1.663e-03 4.417e-02 3.014e-04 2.442e-01 9.665e-01
  5.173e-03 1.658e-02 2.078e-03 5.144e-02]]
```

Listing 7d:Binarize Data

```
In [89]: # 7d. Binarize Data
         binarizer = Binarizer(threshold=0.0).fit(X)
         binaryX = binarizer.transform(X)
         set_printoptions(precision=3)
         print(binaryX[0:5, :])
```

```
[[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]
```

Note that the higher the quality the higher the average alcohol concentration, increased by about 1% at each level. Although lower quality wines have the lowest standard deviation. The chlorides and volatile acidity are less present and presented smaller standard deviation in wines of higher quality. The free sulfur dioxide is higher with higher quality, but their standard deviation decreases with the increase in quality. Higher quality has less fixed acidity, but the standard deviation is slightly higher in mean quality.