In [6]:

```
import sys
import numpy
import matplotlib
import pandas
import sklearn

print('Python: {}'.format(sys.version))
print('numpy: {}'.format(numpy.__version__))
print('matplotlib: {}'.format(matplotlib.__version__))
print('pandas: {}'.format(pandas.__version__))
print('sklearn: {}'.format(sklearn.__version__))
```

```
Python: 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
numpy: 1.18.1
matplotlib: 3.1.3
pandas: 1.0.1
sklearn: 0.22.1
```

In [8]:

```
import numpy as np
from sklearn import preprocessing
from sklearn.model_selection import cross_validate
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn import model_selection
from sklearn.metrics import classification_report, accuracy_score
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
import pandas as pd
```

In [24]:

```
#url="E:\ML\Projects\Breast Cancer Detection\breast-cancer-wisconsin.data"
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsi
n/breast-cancer-wisconsin.data"
names=['id', 'clump_thickness', 'uniform_cell_size', 'uniform_cell_shape',
       'marginal_adhesion', 'single_epithelial_size', 'bare_nuclei',
       'bland_chromatin', 'normal_nucleoli', 'mitoses', 'class']
df = pd.read_csv(url, names=names)
```

In [25]:

```
df.replace('?',-99999,inplace=True)
print(df.axes)
```

```
[RangeIndex(start=0, stop=699, step=1), Index(['id', 'clump_thickness', 'u
niform_cell_size', 'uniform_cell_shape',
       'marginal_adhesion', 'single_epithelial_size', 'bare_nuclei',
       'bland_chromatin', 'normal_nucleoli', 'mitoses', 'class'],
      dtype='object')]
```

In [26]:

```
#print(df)
#print(df.shape)
df.drop(['id'],1,inplace=True)
print(df)
print(df.shape)
```

```
     clump_thickness  uniform_cell_size  uniform_cell_shape  \
0                  5                  1                   1
1                  5                  4                   4
2                  3                  1                   1
3                  6                  8                   8
4                  4                  1                   1
..               ...                ...                 ...
694                3                  1                   1
695                2                  1                   1
696                5                 10                  10
697                4                  8                   6
698                4                  8                   8

     marginal_adhesion  single_epithelial_size bare_nuclei  bland_chromati
n  \
0                    1                       2           1
3
1                    5                       7          10
3
2                    1                       2           2
3
3                    1                       3           4
3
4                    3                       2           1
3
..                 ...                     ...         ...
...
694                  1                       3           2
1
695                  1                       2           1
1
696                  3                       7           3
8
697                  4                       3           4               1
0
698                  5                       4           5               1
0

     normal_nucleoli  mitoses  class
0                  1        1      2
1                  2        1      2
2                  1        1      2
3                  7        1      2
4                  1        1      2
..               ...      ...    ...
694                1        1      2
695                1        1      2
696               10        2      4
697                6        1      4
698                4        1      4

[699 rows x 10 columns]
(699, 10)
```

In [28]:

```
print(df.loc[6])
```

```
clump_thickness            1
uniform_cell_size          1
uniform_cell_shape         1
marginal_adhesion          1
single_epithelial_size     2
bare_nuclei               10
bland_chromatin            3
normal_nucleoli            1
mitoses                    1
class                      2
Name: 6, dtype: object
```

In [29]:

```
print(df.describe())
```

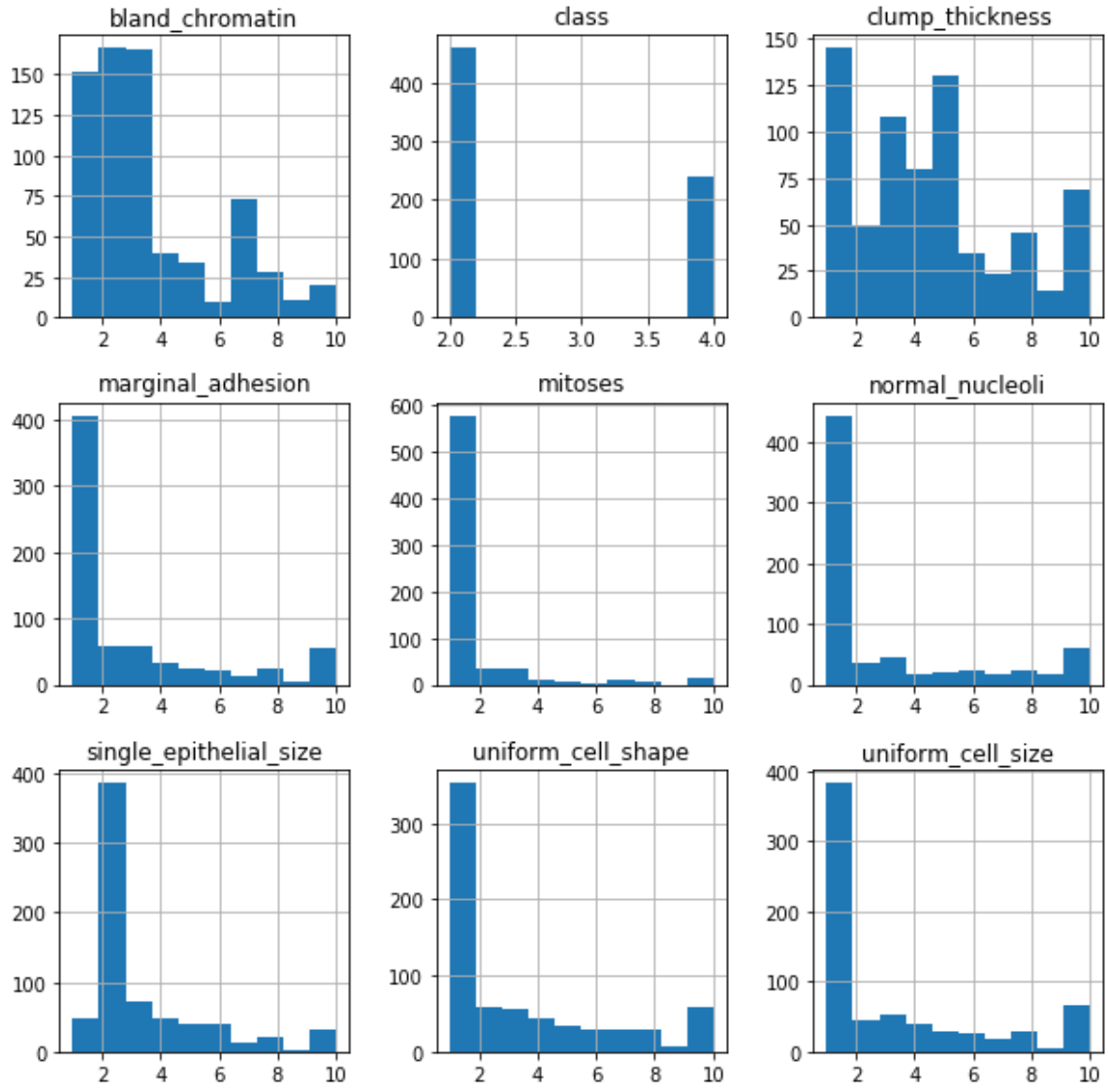
```
       clump_thickness  uniform_cell_size  uniform_cell_shape  \
count       699.000000         699.000000          699.000000
mean          4.417740           3.134478            3.207439
std           2.815741           3.051459            2.971913
min           1.000000           1.000000            1.000000
25%           2.000000           1.000000            1.000000
50%           4.000000           1.000000            1.000000
75%           6.000000           5.000000            5.000000
max          10.000000          10.000000           10.000000

       marginal_adhesion  single_epithelial_size  bland_chromatin  \
count         699.000000              699.000000       699.000000
mean            2.806867                3.216023         3.437768
std             2.855379                2.214300         2.438364
min             1.000000                1.000000         1.000000
25%             1.000000                2.000000         2.000000
50%             1.000000                2.000000         3.000000
75%             4.000000                4.000000         5.000000
max            10.000000               10.000000        10.000000

       normal_nucleoli     mitoses       class
count       699.000000  699.000000  699.000000
mean          2.866953    1.589413    2.689557
std           3.053634    1.715078    0.951273
min           1.000000    1.000000    2.000000
25%           1.000000    1.000000    2.000000
50%           1.000000    1.000000    2.000000
75%           4.000000    1.000000    4.000000
max          10.000000   10.000000    4.000000
```

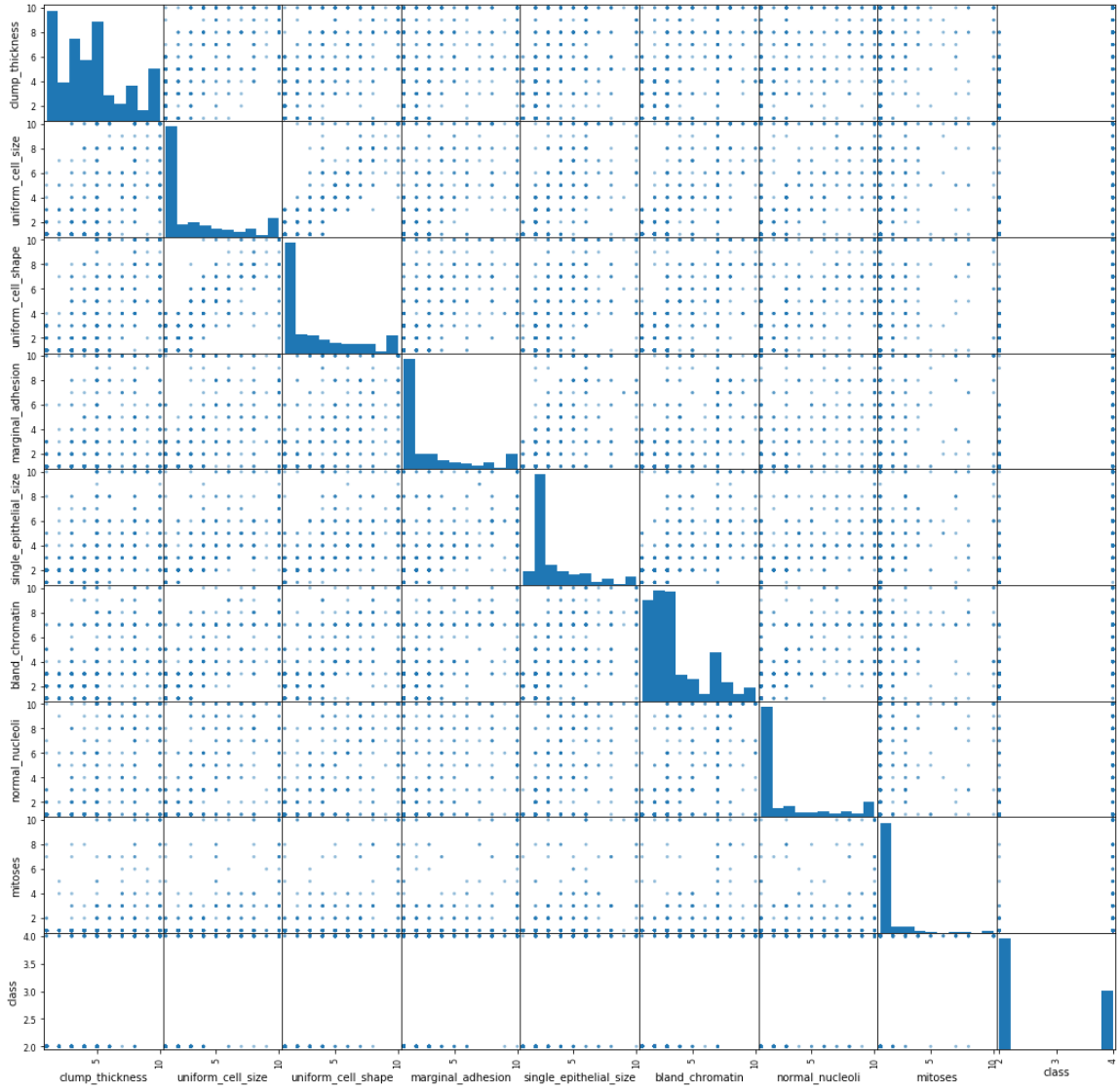
In [31]:

```
df.hist(figsize = (10, 10))
plt.show()
```



In [210]:

```
scatter_matrix(df,figsize=(18,18))
plt.show()
```



In [500]:

```
X=np.array(df.drop(['class'],1))
y=np.array(df['class'])
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size=0.2
)

models=[]
models.append(('KNN',KNeighborsClassifier(n_neighbors=6)))
models.append(('SVM',SVC(kernel='linear')))
#models.append(('SVM',SVC(kernel='rbf')))
#models.append(('SVM',SVC(kernel='poly')))
#models.append(('SVM',SVC(kernel='sigmoid')))

results = []
names = []

for name,model in models:
    kfold=model_selection.KFold(n_splits=10, random_state = 0, shuffle=True)
    cv_results=model_selection.cross_val_score(model,X_train,y_train,cv=kfold,scoring=
'accuracy')
    results.append(cv_results)
    names.append(name)
    print("{}:accuracy->{}(std->{})".format(name, cv_results.mean(), cv_results.std()))
```

```
KNN:accuracy->0.9678571428571427(std->0.022303564279994276)
SVM:accuracy->0.9446103896103896(std->0.03780654211128742)
```

In [501]:

```
for name, model in models:
    model.fit(X_train, y_train)
    predictions = model.predict(X_test)
    print(name)
    print(accuracy_score(y_test, predictions))
    print(classification_report(y_test, predictions))
```

```
KNN
0.9714285714285714
              precision    recall  f1-score   support

           2       0.99      0.97      0.98        97
           4       0.93      0.98      0.95        43

    accuracy                           0.97       140
   macro avg       0.96      0.97      0.97       140
weighted avg       0.97      0.97      0.97       140

SVM
0.95
              precision    recall  f1-score   support

           2       0.99      0.94      0.96        97
           4       0.88      0.98      0.92        43

    accuracy                           0.95       140
   macro avg       0.93      0.96      0.94       140
weighted avg       0.95      0.95      0.95       140
```

In [508]:

```
clf = SVC(kernel='linear')

clf.fit(X_train, y_train)
accuracy = clf.score(X_test, y_test)
print(accuracy)

example_measures = np.array([[4,1,3,5,2,1,4,1,10]])
#print(example_measures)
example_measures = example_measures.reshape(len(example_measures), -1)
#print(example_measures)
prediction = clf.predict(example_measures)
print(prediction)
```

```
0.95
[4]
```