In [3]:

```
%matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split

fruits=pd.read_table('fruit_data_with_colors.txt')
```

In [4]:

```
fruits.head()
```

Out[4]:

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79

In [5]:

```
fruits.shape
```

Out[5]:

```
(59, 7)
```

In [6]:

```
lookup_fruit_name=dict(zip(fruits.fruit_label.unique(), fruits.fruit_name.unique()))
#x =[1,2,3] y=[a,b,c]
# zip (x,y) -> [1,a]. [2,b], [3,a]
```

In [7]:

```
lookup_fruit_name
```

Out[7]:

```
{1: 'apple', 2: 'mandarin', 3: 'orange', 4: 'lemon'}
```

In [8]:

```
from matplotlib import cm
#feature
X=fruits[['height','width','mass','color_score']]
Y=fruits['fruit_label']
X_train, X_test, Y_train, Y_test=train_test_split(X,Y,random_state=0)
#랜덤하게 75%:25%
```

In [9]:

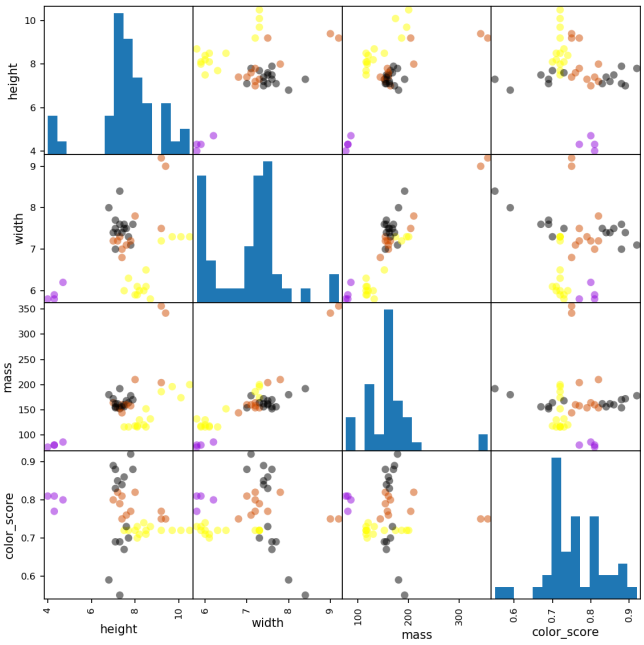
```
X_train.head()
```

Out[9]:

	height	width	mass	color_score
42	7.2	7.2	154	0.82
48	10.1	7.3	174	0.72
7	4.0	5.8	76	0.81
14	7.3	7.6	152	0.69
32	7.0	7.2	164	0.80

In [11]:

```
cmap=cm.get_cmap('gnuplot')
scatter=pd.plotting.scatter_matrix(X_train,c=Y_train,marker='o',s=40,hist_kwds={'bins':15},figsi
ze=(9,9),cmap=cmap)#다차원
```



In [12]:

```
from matplotlib import cm
#feature
X=fruits[['mass','width','height']]
Y=fruits['fruit_label']
X_train, X_test, Y_train, Y_test=train_test_split(X,Y,random_state=0)
#랜덤하게 75%:25%
```

In [14]:

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train,Y_train)
#학습
```

Out[14]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

In [15]:

```
knn.score(X_test,Y_test) #정확도
```

Out[15]:

```
0.5333333333333333
```

In [16]:

```
knn.score(X_train,Y_train) #정확도
```

Out[16]:

```
0.7954545454545454
```

In [17]:

```
fruit_prediction=knn.predict([[20,4.3,5.5]])
lookup_fruit_name[fruit_prediction[0]]
```

Out[17]:

```
'mandarin'
```

In [19]:

```
#parameter change
k_range=range(1,20)
scores=[]

for k in k_range:
    knn=KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train,Y_train)
    scores.append(knn.score(X_test,Y_test))
```

In [20]:

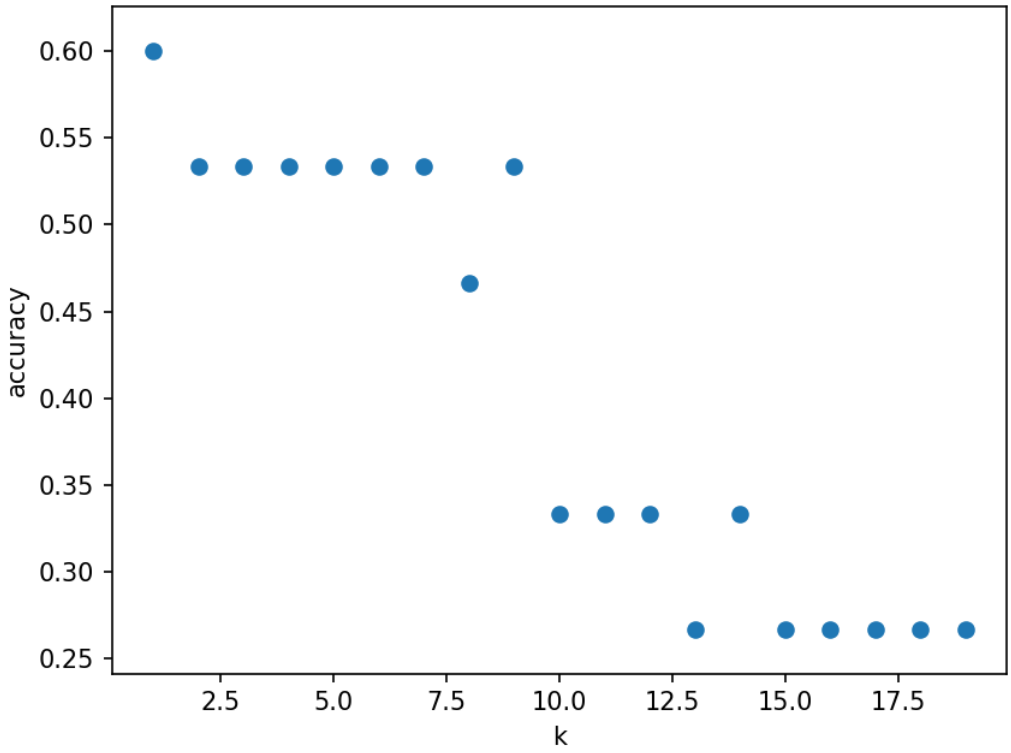
```
scores
```

Out[20]:

```
[0.6,
 0.5333333333333333,
 0.5333333333333333,
 0.5333333333333333,
 0.5333333333333333,
 0.5333333333333333,
 0.5333333333333333,
 0.4666666666666667,
 0.5333333333333333,
 0.3333333333333333,
 0.3333333333333333,
 0.3333333333333333,
 0.26666666666666666,
 0.3333333333333333,
 0.26666666666666666,
 0.26666666666666666,
 0.26666666666666666,
 0.26666666666666666,
 0.26666666666666666]
```

In [21]:

```
plt.figure()
plt.xlabel('k')
plt.ylabel('accuracy')
plt.scatter(k_range,scores)
```



Out[21]:

```
<matplotlib.collections.PathCollection at 0x27be79d5688>
```

In [22]:

```
# Sensitiveness of K-NN classification according to train/test split proportion

t=[0.8,0.7,0.6,0.5,0.4,0.3,0.2]
knn=KNeighborsClassifier(n_neighbors=5)

scores=[]
for s in t:
    X_train, X_test, Y_train, Y_test=train_test_split(X,Y,test_size=1-s)
    knn.fit(X_train,Y_train)
    scores.append(knn.score(X_test,Y_test))
```

In [23]:

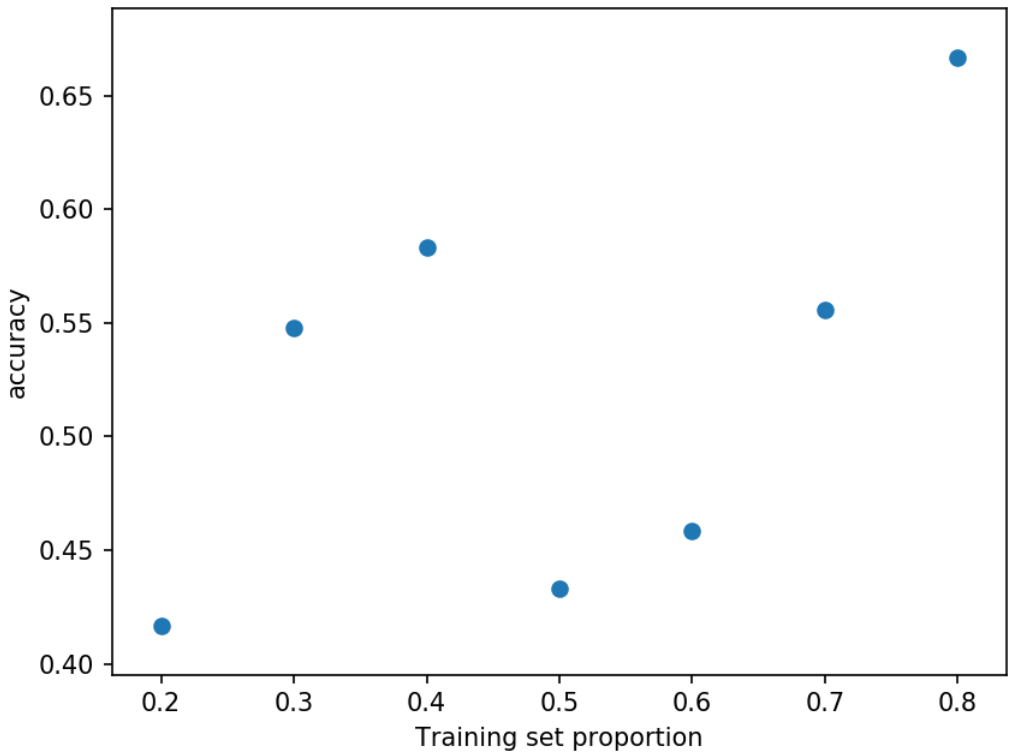
```
scores
```

Out[23]:

```
[0.6666666666666666,
 0.5555555555555556,
 0.4583333333333333,
 0.43333333333333335,
 0.5833333333333334,
 0.5476190476190477,
 0.4166666666666667]
```

In [26]:

```
plt.figure()
plt.xlabel('Training set proportion')
plt.ylabel('accuracy')
plt.scatter(t, scores)
```



Out[26]:

```
<matplotlib.collections.PathCollection at 0x27be7b99e88>
```

In [ ]:

```

```