In [1]:

```
#Loading the Library

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cross_validation import train_test_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy_score

from sklearn.cross_validation import cross_val_score

from matplotlib.colors import ListedColormap

from sklearn import neighbors
```

```
C:\Users\ajant\python\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarning: This modul
e was deprecated in version 0.18 in favor of the model_selection module into which all the refactored c
lasses and functions are moved. Also note that the interface of the new CV iterators are different from
that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

In [5]:

```
#loading the Dataset

#loading the Dataset
dataset = pd.read_csv('D:/Machine Learning Datasets/Iris.csv')
dataset
```

Out[5]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa
12	4.8	3.0	1.4	0.1	Iris-setosa
13	4.3	3.0	1.1	0.1	Iris-setosa
14	5.8	4.0	1.2	0.2	Iris-setosa
15	5.7	4.4	1.5	0.4	Iris-setosa
16	5.4	3.9	1.3	0.4	Iris-setosa
17	5.1	3.5	1.4	0.3	Iris-setosa
18	5.7	3.8	1.7	0.3	Iris-setosa
19	5.1	3.8	1.5	0.3	Iris-setosa

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
20	5.4	3.4	1.7	0.2	Iris-setosa
21	5.1	3.7	1.5	0.4	Iris-setosa
22	4.6	3.6	1.0	0.2	Iris-setosa
23	5.1	3.3	1.7	0.5	Iris-setosa
24	4.8	3.4	1.9	0.2	Iris-setosa
25	5.0	3.0	1.6	0.2	Iris-setosa
26	5.0	3.4	1.6	0.4	Iris-setosa
27	5.2	3.5	1.5	0.2	Iris-setosa
28	5.2	3.4	1.4	0.2	Iris-setosa
29	4.7	3.2	1.6	0.2	Iris-setosa
...	...	...	...	...	...
120	6.9	3.2	5.7	2.3	Iris-virginica
121	5.6	2.8	4.9	2.0	Iris-virginica
122	7.7	2.8	6.7	2.0	Iris-virginica
123	6.3	2.7	4.9	1.8	Iris-virginica
124	6.7	3.3	5.7	2.1	Iris-virginica
125	7.2	3.2	6.0	1.8	Iris-virginica
126	6.2	2.8	4.8	1.8	Iris-virginica
127	6.1	3.0	4.9	1.8	Iris-virginica
128	6.4	2.8	5.6	2.1	Iris-virginica
129	7.2	3.0	5.8	1.6	Iris-virginica
130	7.4	2.8	6.1	1.9	Iris-virginica
131	7.9	3.8	6.4	2.0	Iris-virginica
132	6.4	2.8	5.6	2.2	Iris-virginica
133	6.3	2.8	5.1	1.5	Iris-virginica
134	6.1	2.6	5.6	1.4	Iris-virginica
135	7.7	3.0	6.1	2.3	Iris-virginica
136	6.3	3.4	5.6	2.4	Iris-virginica
137	6.4	3.1	5.5	1.8	Iris-virginica
138	6.0	3.0	4.8	1.8	Iris-virginica
139	6.9	3.1	5.4	2.1	Iris-virginica
140	6.7	3.1	5.6	2.4	Iris-virginica
141	6.9	3.1	5.1	2.3	Iris-virginica
142	5.8	2.7	5.1	1.9	Iris-virginica
143	6.8	3.2	5.9	2.3	Iris-virginica
144	6.7	3.3	5.7	2.5	Iris-virginica
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [15]:

```
#create X (features) and y (response)

array=dataset.values
```

```
array=dataset.values

x = array[:,0:3]

y = array[:,4]
```

In [16]:

```
#Splitting the dataset into the Training set and Test set

from sklearn.cross_validation import train_test_split

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.33, random_state = 0)
```

In [17]:

```
#Fitting classifier to the Training set
from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier (n_neighbors=5 ,metric='minkowski', p=2 )

classifier.fit(X_train , y_train)
```

Out[17]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
           weights='uniform')
```

In [18]:

```
#Predict the response
pred = classifier.predict(X_test)
```

In [19]:

```
#Evaluate accuracy
acc = accuracy_score(y_test, pred) * 100

print('\nThe accuracy of the knn classifier for k = 5 is %d%%' % acc)
```

```
The accuracy of the knn classifier for k = 5 is 98%
```

In [20]:

```
#Creating odd list of K for KNN
myList = list(range(0,50))

neighbors = list(filter(lambda x: x % 2 != 0, myList))
```

In [24]:

```
#Empty list that will hold cv scores

cv_scores = []

#Performing 10-fold cross-validation with K=5 for KNN (the n_neighbors parameter)
#k = 5 for KNeighborsClassifier

#scoring='accuracy' for evaluation metric


knn=KNeighborsClassifier(n_neighbors=5)

scores=cross_val_score(knn,x,y,cv=10,scoring='accuracy')

print (scores)

#In the first iteration, the accuracy is 100%

#Second iteration, the accuracy is 93% and so on
```

```
[ 1.          0.93333333  1.          1.          0.86666667  0.8
  0.93333333  0.93333333  1.          1.        ]
```

In [26]:

```
#performing 10-fold cross validation
for k in neighbors:
```

```
    knn = KNeighborsClassifier(n_neighbors=k)

    scores = cross_val_score(knn, X_train, y_train, cv=10, scoring='accuracy')

    cv_scores.append(scores.mean())

    print (cv_scores)
```

```
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
```

7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.9409090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.9016161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333

```
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90
161616161616165]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90
161616161616165, 0.88696969696969696]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90
161616161616165, 0.88696969696969696, 0.89530303030303027]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90
161616161616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90
161616161616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134]
[0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333333333333
7, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090909090909
103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309090909090
9091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90161616161
616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.886212121
21212134, 0.92070707070707081, 0.94722222222222219, 0.95833333333333326, 0.96944444444444444, 0.9583333
3333333337, 0.94722222222222219, 0.94924242424242422, 0.94090909090909103, 0.94090909090909103, 0.94090
909090909103, 0.9520202020202021, 0.94090909090909103, 0.94090909090909103, 0.94090909090909103, 0.9309
0909090909091, 0.93090909090909091, 0.91348484848484846, 0.90363636363636357, 0.90161616161616165, 0.90
161616161616165, 0.88696969696969696, 0.89530303030303027, 0.90363636363636368, 0.88621212121212134, 0.
88621212121212134]
```

In [30]:

```
#Changing to misclassification error


MSE = [1 - x for x in cv_scores]
MSE
```

Out[30]:

```
[0.079292929292929193,
 0.052777777777777812,
 0.041666666666666741,
 0.030555555555555558,
 0.04166666666666663,
 0.052777777777777812,
 0.050757575757575779,
```

```
 0.059090909090908972,
 0.059090909090908972,
 0.059090909090908972,
 0.0479797979797979,
 0.059090909090908972,
 0.059090909090908972,
 0.059090909090908972,
 0.069090909090909092,
 0.069090909090909092,
 0.086515151515151545,
 0.096363636363636429,
 0.098383838383838351,
 0.098383838383838351,
 0.11303030303030304,
 0.10469696969696973,
 0.096363636363636318,
 0.11378787878787866,
 0.11378787878787866,
 0.079292929292929193,
 0.052777777777777812,
 0.041666666666666741,
 0.030555555555555558,
 0.04166666666666663,
 0.052777777777777812,
 0.050757575757575779,
 0.059090909090908972,
 0.059090909090908972,
 0.059090909090908972,
 0.0479797979797979,
 0.059090909090908972,
 0.059090909090908972,
 0.059090909090908972,
 0.069090909090909092,
 0.069090909090909092,
 0.086515151515151545,
 0.096363636363636429,
 0.098383838383838351,
 0.098383838383838351,
 0.11303030303030304,
 0.10469696969696973,
 0.096363636363636318,
 0.11378787878787866,
 0.11378787878787866]
```

In [44]:

```
# determining best k
optimal_k = neighbors[MSE.index(min(MSE))]
print('\nThe optimal number of neighbors is %d.' % optimal_k)
```

```
The optimal number of neighbors is 7.
```

In [49]:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn import neighbors, datasets
n_neighbors = 5
# import some data to play with
iris = datasets.load_iris()
# we only take the first two features. We could avoid this ugly
# slicing by using a two-dim dataset
X = iris.data[:, :2]
y = iris.target
h = .02 # step size in the mesh
# Create color maps
cmap_light = ListedColormap(['#FFAAAA', '#AAFFAA', '#AAAAFF'])
cmap_bold = ListedColormap(['#FF0000', '#00FF00', '#0000FF'])
for weights in ['uniform']:
 # we create an instance of Neighbours Classifier and fit the data.
 clf = neighbors.KNeighborsClassifier(n_neighbors, weights=weights)
 clf.fit(X, y)
 # Plot the decision boundary. For that, we will assign a color to each
 # point in the mesh [x_min, x_max]x[y_min, y_max].
 x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
 y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
```

```
 xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
 np.arange(y_min, y_max, h))
 Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
 # Put the result into a color plot
 Z = Z.reshape(xx.shape)
 plt.figure()
 plt.pcolormesh(xx, yy, Z, cmap=cmap_light)
 # Plot also the training points
 plt.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap_bold,
 edgecolor='k', s=20)
 plt.xlim(xx.min(), xx.max())
 plt.ylim(yy.min(), yy.max())
 plt.title("KNN")
 plt.xlabel("SepalLengthCm")
 plt.ylabel("SepalWidthCm")
plt.show()
```