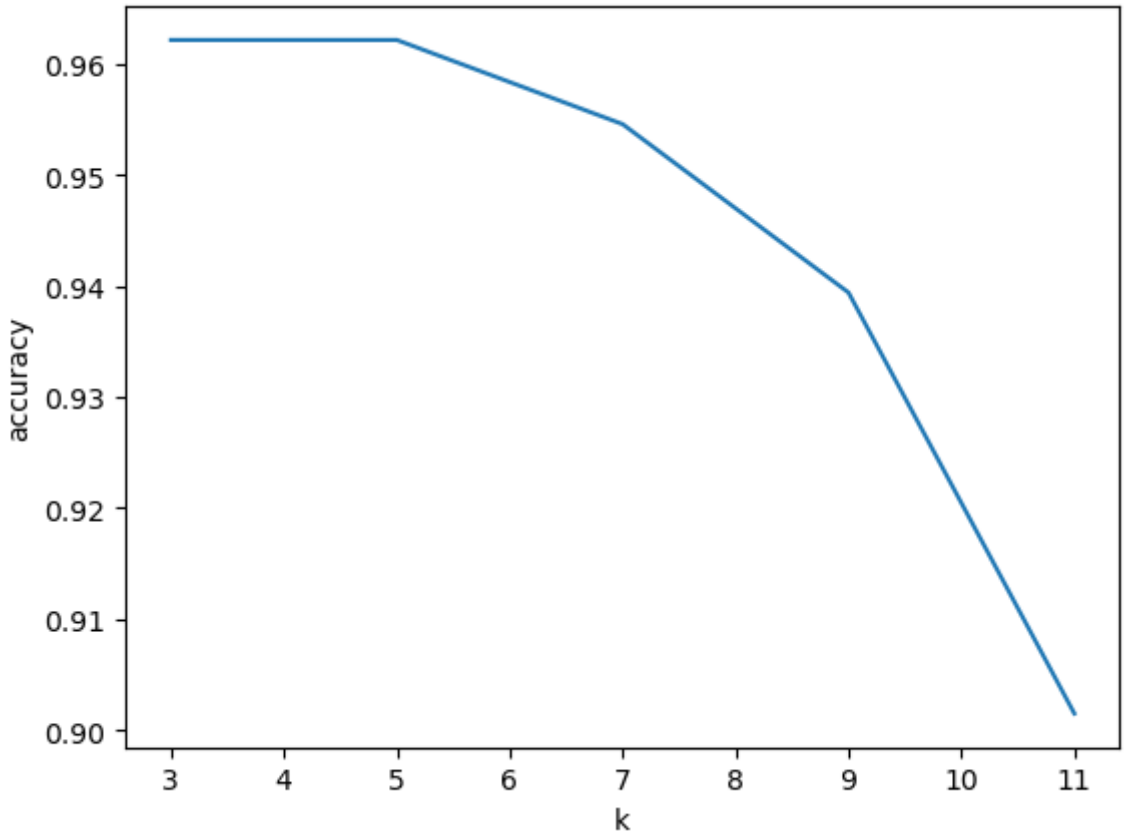
In [97]:
```
import pandas as pd
import numpy as np
import sklearn
import matplotlib.pyplot as plt
from sklearn.model_selection \
import train_test_split
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler , LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

Quetion#1. take k = 3,5,7,9,11. For each value of k compute the accuracy of your k-NN classifier on year 1 data. On x axis you plot k and on y-axis you plot accuracy. What is the optimal value of k for year 1?

In [98]:
```
df=pd.read_csv("NVDA_weekly_return_volatility.csv")
year=df['Year'].unique()
Q1_label=[]
yearly_mean=df.groupby('Year')['mean_return'].mean().values
for i in range(len(year)):
    for j in range(len(df)):
        if df['Year'][j]==year[i] and df["mean_return"][j]>yearly_mean[i]:
            Q1_label.append('green')
        elif df['Year'][j]==year[i]:
            Q1_label.append('red')
df['label']=Q1_label

def Q1_KNN(Q1_data,n):
    Q1_X=df[["mean_return","volatility"]]
    Q1_y=df["label"]
    Q1_X_train, Q1_X_test, Q1_y_train, Q1_y_test = train_test_split(Q1_X, Q1_y,
    knn = KNeighborsClassifier(n_neighbors=n)
    knn.fit(Q1_X_train, Q1_y_train)
    return accuracy_score(Q1_y_test, knn.predict(Q1_X_test))
df_q1_2017=df.loc[df['Year']==2017]
df_q1_2018=df.loc[df['Year']==2018]
k=[3,5,7,9,11]
q1_acc=[]
for i in k:
    res=Q1_KNN(df_q1_2017,i)
    q1_acc.append(res)
    print("if k=",i,'the accurcy is',res)
plt.plot(k,q1_acc)
plt.ylabel("accuracy")
plt.xlabel("k")
plt.show()
```

```
if k= 3 the accurcy is 0.9621212121212122
if k= 5 the accurcy is 0.9621212121212122
if k= 7 the accurcy is 0.9545454545454546
if k= 9 the accurcy is 0.9393939393939394
if k= 11 the accurcy is 0.9015151515151515
```



Question#2. use the optimal value of k from year 1 to predict labels for year 2. What is your accuracy?

In [99]:
```
Q2_X=df_q1_2017[["mean_return","volatility"]]
Q2_y=df_q1_2017["label"]
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(Q2_X, Q2_y)
Q2_acc=accuracy_score(df_q1_2018['label'], knn.predict(df_q1_2018[["mean_returr
print('the accucry is',Q2_acc)
```

```
the accucry is 0.9245283018867925
```

Question#3. using the optimal value for k from year 1, compute the confusion matrix for year 2

In [100…
```
a= confusion_matrix(df_q1_2018['label'], knn.predict(df_q1_2018[["mean_return",
print('the confusion matrix is\n',a)
```

```
the confusion matrix is
 [[25  4]
 [ 0 24]]
```

Question#4. what is true positive rate (sensitivity or recall) and true negative rate (specificity) for year 2?

In [101…
```
Q4_TN, Q4_FP, Q4_FN, Q4_TP = confusion_matrix(df_q1_2018['label'], knn.predict(
Q4_TPR=Q4_TP/(Q4_TP+Q4_FN)
Q4_TNR=Q4_TN/(Q4_TN+Q4_FP)
print('true positive rate',Q4_TPR)
print('true negative rate',Q4_TNR)
```

```
true positive rate 1.0
true negative rate 0.8620689655172413
```

Question#5. implement a trading strategy based on your labels for year 2 and compare the performance with the ″buy-and-hold″ strategy. Which strategy results in a larger amount at the end of the year?

In [ ]: