In [10]:
```
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
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder
import warnings
warnings.filterwarnings('ignore')
```

In [11]:
```
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
Q1_X=df[df["Year"]==2017][["mean_return","volatility"]]
Q1_y=df[df["Year"]==2017]["label"]
Q2_X=df.loc[df["Year"]==2018][["mean_return","volatility"]]
Q2_y=df.loc[df["Year"]==2018]["label"]
```

1. implement a decision tree and compute its accuracy for year 2

In [12]:
```
from sklearn import tree
clf = tree.DecisionTreeClassifier(criterion = 'entropy')
clf = clf.fit(Q1_X, Q1_y)
#implement a decision tree and compute its accuracy for year 2
y_pred=clf.predict(Q2_X)
accuracy_score(Q2_y,y_pred)
```

Out[12]:
```
0.9811320754716981
```

1. compute the confusion matrix for year 2

In [13]:
```
print("the matrix for year 2 is :\n",confusion_matrix(Q2_y,y_pred))
```

```
the matrix for year 2 is :
 [[28  1]
 [ 0 24]]
```

1. what is true positive rate and true negative rate for year 2?

In [14]:
```
# what is true positive rate and true negative rate for year 2?
# true positive rate
TPR=confusion_matrix(Q2_y,y_pred)[0][0]/(confusion_matrix(Q2_y,y_pred)[0][0]+co
print("the true positive rate for year 2 is :",TPR)
# true negative rate
TNR=confusion_matrix(Q2_y,y_pred)[1][1]/(confusion_matrix(Q2_y,y_pred)[1][0]+co
print("the true negative rate for year 2 is :",TNR)
```

```
the true positive rate for year 2 is : 0.9655172413793104
the true negative rate for year 2 is : 1.0
```

1. 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 [16]:
```
#implement a trading strategy based on your labels for year 2 and compare the p
# buy and hold strategy
buy_and_hold=100
for i in range(len(Q2_y)):
    if Q2_y.iloc[i]=="green":
        buy_and_hold=buy_and_hold*(1+df["mean_return"][i])
print("the amount of buy and hold strategy is :",buy_and_hold)
# trading strategy
trading_strategy=100
for i in range(len(y_pred)):
    if y_pred[i]=="green":
        trading_strategy=trading_strategy*(1+df["mean_return"][i])
print("the amount of trading strategy is :",trading_strategy)
```

```
the amount of buy and hold strategy is : 11007.201951211806
the amount of trading strategy is : 10825.868651302488
```