In [73]:
```
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.naive_bayes import GaussianNB
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.preprocessing import StandardScaler
```

In [74]:
```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

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"]
lda_classifier = LDA().fit(Q1_X, Q1_y)
lda_prediction = lda_classifier.predict(Q2_X)
lda_error_rate = np.mean(lda_prediction != Q2_y)
print(lda_error_rate)
```

```
0.16981132075471697
```

In [75]:
```
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis as QDA
qda_classifier = QDA().fit(Q1_X, Q1_y)
qda_prediction = qda_classifier.predict(Q2_X)
qda_error_rate = np.mean(qda_prediction != Q2_y)
print(qda_error_rate)
```

```
0.1320754716981132
```

1. what is the equation for linear and quadratic classifier found from year 1 data?

In [76]:
```
print("for linear coef and intercept",lda_classifier.coef_,lda_classifier.inter
print("so for linear the function is y=-10.63x^2+1.0039x+1.406")

print("for Quadratic classifier, we can't get the corf_ and intercet")
```

```
for linear coef and intercept [[-10.62650115   1.00391457]] [1.40608655]
so for linear the function is y=-10.63x^2+1.0039x+1.406
for Quadratic classifier, we can't get the corf_ and intercet
```

1. what is the accuracy for year 2 for each classifier. Which classifier is "better"?

In [77]:
```
print("the year2 accuracy for lda is",accuracy_score(Q2_y, lda_classifier.predi
print("the year2 accuracy for qda is",accuracy_score(Q2_y, qda_classifier.predi
```

```
the year2 accuracy for lda is 0.8301886792452831
the year2 accuracy for qda is 0.8679245283018868
```

1. compute the confusion matrix for year 2 for each classifier

In [78]:
```
lda= confusion_matrix(Q2_y, lda_classifier.predict(Q2_X))
qda= confusion_matrix(Q2_y, qda_classifier.predict(Q2_X))
print('the confusion matrix for lda is\n',lda)
print('the confusion matrix for qda is\n',qda)
```

```
the confusion matrix for lda is
 [[20  9]
 [ 0 24]]
the confusion matrix for qda is
 [[22  7]
 [ 0 24]]
```

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

In [79]:
```
Q4_TN, Q4_FP, Q4_FN, Q4_TP = confusion_matrix(Q2_y, lda_classifier.predict(Q2_X
Q4_TPR=Q4_TP/(Q4_TP+Q4_FN)
Q4_TNR=Q4_TN/(Q4_TN+Q4_FP)
print('true positive rate for LDA',Q4_TPR)
print('true negative rate for LDA',Q4_TNR)
Q4_TN, Q4_FP, Q4_FN, Q4_TP = confusion_matrix(Q2_y, qda_classifier.predict(Q2_X
Q4_TPR=Q4_TP/(Q4_TP+Q4_FN)
Q4_TNR=Q4_TN/(Q4_TN+Q4_FP)
print('true positive rate for QDA',Q4_TPR)
print('true negative rate for QDA',Q4_TNR)
```

```
true positive rate 1.0
true negative rate 0.6896551724137931
true positive rate 1.0
true negative rate 0.7586206896551724
```

1. implement trading strategyies based on your labels for year 2 (for both linear and quadratic) and compare the perfor- mance with the "buy-and-hold" strategy. Which strategy results in a larger amount at the end of the year?

In [ ]:
```
```