In [303]:
```
import pandas as pd
import statsmodels.api as sm
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.formula.api import ols
from sklearn import tree
from sklearn.model_selection import GridSearchCV
import graphviz
```

In [232]:
```
party_control_org = pd.read_csv("C:\\Users\\shawn\\Desktop\\MIT college work\\17.

start = int(party_control_org["Year"][0][:4])
end = int(party_control_org["Year"][len(party_control_org["Year"])-1][:4])
```

In [233]:
```
party_control_list = []

for i in range(start,end):
    index = (i - start)//2
    pres = party_control_org["President"][index]
    senate = party_control_org["Senate"][index]
    house = party_control_org["House"][index]
    if pres == senate and pres == house:
        gov  = "Unified"
        gov_marker = 1
        if pres == "R":
            party_marker = 1
        else:
            party_marker = 0
    else:
        gov = "Divided"
        gov_marker = 0
        party_marker = None
    party_control_list.append([i,pres,senate, house, gov, gov_marker, party_marke

party_control = pd.DataFrame(party_control_list,columns=["Year", "President", "Se
```

In [293]:
```
sp_returns = pd.read_csv("C:\\Users\\shawn\\Desktop\\MIT college work\\17.801\\Fi
df_congress = pd.read_csv("C:\\Users\\shawn\\Desktop\\MIT college work\\17.801\\F
df_congress_year = df_congress[['Year', 'True Approval']].groupby(['Year']).mean(
df_debt = pd.read_csv("C:\\Users\\shawn\\Desktop\\MIT college work\\17.801\\Final
df_military = pd.read_csv("C:\\Users\\shawn\\Desktop\\MIT college work\\17.801\\F
df_inflation = pd.read_csv("C:\\Users\\shawn\\Desktop\\MIT college work\\17.801\\
df_gdp =  pd.read_csv("C:\\Users\\shawn\\Desktop\\MIT college work\\17.801\\Final
print(df_gdp)

df_inflation["Change in CPI Index Percent"] = df_inflation["Change in CPI Index P
df_inflation = df_inflation.dropna()
df_inflation['Year'] = df_inflation['Year'].astype(int)
```

	Year	GDP Growth	Real GDP
0	2018	2.52	18.78
1	2017	2.80	18.32
2	2016	2.03	17.82
3	2015	1.90	17.47
4	2014	2.88	17.14
5	2013	2.61	16.66
6	2012	1.47	16.24
7	2011	1.61	16.00
8	2010	2.57	15.75
9	2009	0.18	15.36
10	2008	-2.75	15.33
11	2007	1.97	15.76
12	2006	2.59	15.46
13	2005	3.13	15.07
14	2004	3.28	14.61
15	2003	4.33	14.15
16	2002	2.09	13.56
17	2001	0.15	13.28
18	2000	2.97	13.26
19	1999	4.81	12.88
20	1998	4.88	12.29
21	1997	4.49	11.72
22	1996	4.42	11.21
23	1995	2.20	10.74
24	1994	4.12	10.51
25	1993	2.61	10.09
26	1992	4.38	9.83
27	1991	1.17	9.42
28	1990	0.60	9.31
29	1989	2.74	9.26
..	...	...	...
59	1959	4.59	3.20
60	1958	2.66	3.06
61	1957	0.35	2.98
62	1956	2.00	2.97
63	1955	6.58	2.91
64	1954	2.73	2.74
65	1953	0.52	2.66
66	1952	5.37	2.65
67	1951	5.47	2.51
68	1950	13.37	2.38
69	1949	-1.53	2.10

```
70  1948        3.89      2.13
71  1947       -0.16      2.06
72  1946      -11.60      2.06
73  1945       -0.98      2.33
74  1944        7.95      2.35
75  1943       17.02      2.18
76  1942       18.88      1.86
77  1941       17.71      1.57
78  1940        8.82      1.33
79  1939        8.02      1.22
80  1938       -3.31      1.13
81  1937        5.12      1.17
82  1936       12.89      1.11
83  1935        8.90      0.99
84  1934       10.80      0.91
85  1933       -1.23      0.82
86  1932      -12.89      0.83
87  1931       -6.41      0.95
88  1930       -8.50      1.02

[89 rows x 3 columns]
```

In [294]:

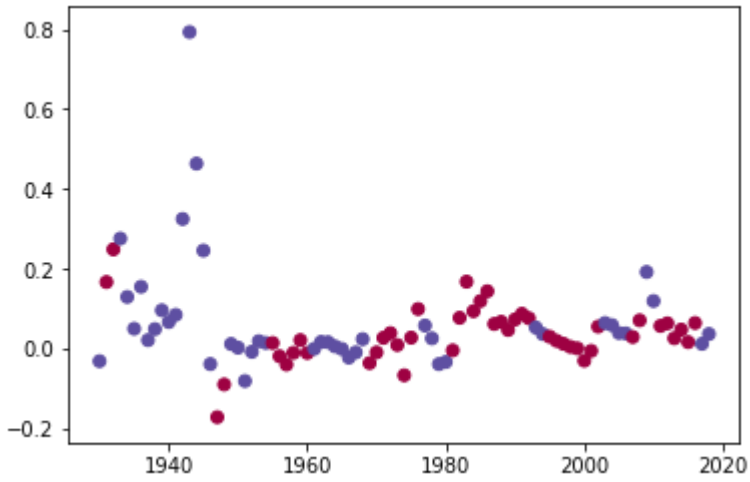
```
df_data = pd.merge(party_control,sp_returns, how= 'outer', on = "Year")
df_data = pd.merge(df_data, df_congress_year, how = 'outer', on = "Year")
df_data = pd.merge(df_data, df_debt, how = 'outer', on = 'Year')
df_data = pd.merge(df_data, df_military, how = 'outer', on = 'Year')
df_data = pd.merge(df_data, df_inflation, how = 'outer', on = 'Year')
df_data = pd.merge(df_data, df_gdp, how = 'outer', on = 'Year')
df_data["Change in True Approval"] = df_data["True Approval"].diff()
df_data['Debt/GDP Ratio'] = df_data['Debt/GDP Ratio'].str.rstrip('%').astype('flo
df_data["Debt"] = df_data["Debt"].replace('[\$,]', '', regex=True).astype(float)
df_data["Military Total"] = df_data["Military Total"].str.replace(',','').astype(
df_data["Change in Debt"], df_data["Change in Debt/GDP Ratio"], df_data["Change i
    df_data["Debt"].diff(), df_data["Debt/GDP Ratio"].diff(), df_data["Debt"].pct
df_data['President'], df_data['House'], df_data['Senate'] = df_data['President']+
df_data['Adjusted Debt'] = df_data['Annual Average CPI Index'][112]*df_data["Debt
df_data["Change in Adjusted Debt Percent"], df_data['Change in Adjusted Debt'] =
df_data["Change in Real GDP"] = df_data["Real GDP"].diff()
```

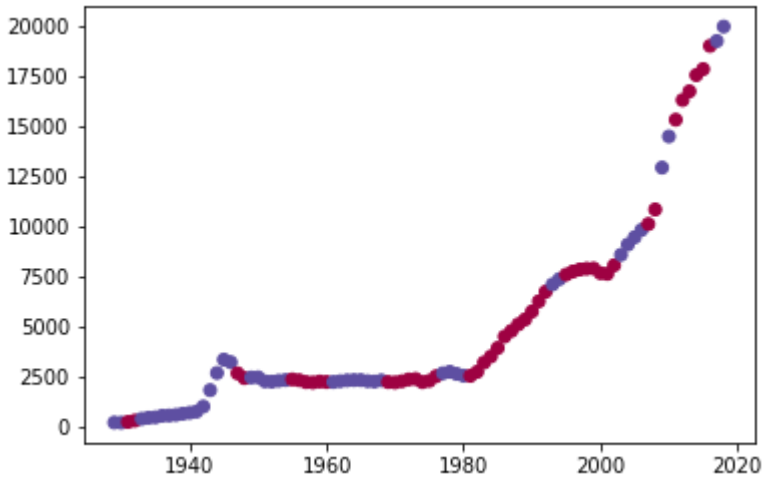
In [295]:
```
from IPython.display import display
pd.options.display.max_columns = None
display(df_data)

df_data.to_csv("C:\\Users\\shawn\\Desktop\\MIT college work\\17.801\\Final Projec
```

NaN	NaN	NaN	NaN	16.8	-6.2	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	17.1	1.8	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	17.1	0.4	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	17.5	2.4	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	17.7	0.9	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	17.4	-1.9	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	17.2	-1.2	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	17.2	0.0	NaN	NaN	NaN	NaN	NaN	NaN	

In [288]:
```
def scatter(y, x, marker):
    fig, ax = plt.subplots()
    scatter = ax.scatter(x, y, c=marker, cmap="Spectral")
    plt.show()
```

In [289]:
```
scatter(df_data["Change in Adjusted Debt Percent"], df_data["Year"], df_data["Gov
scatter(df_data["Adjusted Debt"], df_data["Year"], df_data["Gov Marker"])
```





In [290]:
```
def lin_reg_categorical(model_name):
    fitted_model = model_name.fit()
    print(fitted_model.summary())
```

In [263]:
```
model_econ_parties = ols('Q("S&P Returns") ~ President+ Senate+ House', data = df
model_delta_approval_parties = ols('Q("Change in True Approval") ~ C(President)+ (
model_approval_parties = ols('Q("True Approval") ~ C(President)+ C(Senate)+ C(Hou
model_debt_parties = ols('Debt ~ C(President)+ C(Senate)+ C(House) +Year', data =
model_delta_debt_parties = ols('Q("Change in Debt") ~ C(President)+ C(Senate)+ C(
model_delta_percent_debt_parties = ols('Q("Change in Debt Percent") ~ C(President
model_delta_debt_gdp_parties = ols('Q("Change in Debt/GDP Ratio") ~ President+ Se
```

In [10]:
```
lin_reg_categorical(model_delta_debt_parties) #decent
lin_reg_categorical(model_debt_parties) #really good
lin_reg_categorical(model_delta_percent_debt_parties)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     Q("Change in Debt")   R-squared:                       0.534
Model:                             OLS   Adj. R-squared:                  0.511
Method:                  Least Squares   F-statistic:                     24.03
Date:                 Sat, 16 Nov 2019   Prob (F-statistic):           2.86e-13
Time:                         02:24:57   Log-Likelihood:                -625.56
No. Observations:                   89   AIC:                             1261.
Df Residuals:                       84   BIC:                             1274.
Df Model:                            4
Covariance Type:             nonrobust
================================================================================
===============
                                   coef    std err          t      P>|t|      [0.
025      0.975]
--------------------------------------------------------------------------------
---------------
Intercept                   -2.219e+04   2705.273     -8.204      0.000   -2.76e
+04   -1.68e+04
C(President)[T.RPresident]   -92.0910     64.894     -1.419      0.160    -221.
139      36.957
C(Senate)[T.RSenate]         -81.0037     87.842     -0.922      0.359    -255.
688      93.680
C(House)[T.RHouse]            47.6350     98.786      0.482      0.631    -148.
812     244.082
Year                          11.3923      1.379      8.262      0.000       8.
650      14.134
==============================================================================
Omnibus:                       40.206   Durbin-Watson:                   0.680
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              106.347
Skew:                           1.574   Prob(JB):                     8.07e-24
Kurtosis:                       7.332   Cond. No.                     1.79e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
[2] The condition number is large, 1.79e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
                            OLS Regression Results
==============================================================================
Dep. Variable:                   Debt   R-squared:                       0.719
Model:                            OLS   Adj. R-squared:                  0.706
Method:                 Least Squares   F-statistic:                     54.33
Date:                Sat, 16 Nov 2019   Prob (F-statistic):           1.20e-22
Time:                        02:24:57   Log-Likelihood:                -844.80
No. Observations:                  90   AIC:                             1700.
Df Residuals:                      85   BIC:                             1712.
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
===============
```

```
                                 coef    std err          t      P>|t|      [0.
025      0.975]
------------------------------------------------------------------------------
---------------
Intercept                  -2.771e+05   2.71e+04    -10.209      0.000   -3.31e
+05   -2.23e+05
C(President)[T.RPresident] -1303.1417    675.531     -1.929      0.057   -2646.
279      39.996
C(Senate)[T.RSenate]       -1118.5793    927.952     -1.205      0.231   -2963.
597     726.438
C(House)[T.RHouse]          4107.5996   1013.208      4.054      0.000    2093.
072    6122.128
Year                         142.0533     13.826     10.274      0.000     114.
563     169.543
==============================================================================
Omnibus:                       20.101   Durbin-Watson:                   0.148
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               26.267
Skew:                           1.072   Prob(JB):                     1.98e-06
Kurtosis:                       4.552   Cond. No.                     1.71e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
[2] The condition number is large, 1.71e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
                            OLS Regression Results
==============================================================================
========
Dep. Variable:     Q("Change in Debt Percent")   R-squared:
0.078
Model:                                     OLS   Adj. R-squared:
0.034
Method:                          Least Squares   F-statistic:
1.776
Date:                         Sat, 16 Nov 2019   Prob (F-statistic):
0.141
Time:                                 02:24:57   Log-Likelihood:
64.961
No. Observations:                           89   AIC:
-119.9
Df Residuals:                               84   BIC:
-107.5
Df Model:                                    4
Covariance Type:                     nonrobust
==============================================================================
===============
                                 coef    std err          t      P>|t|      [0.
025      0.975]
------------------------------------------------------------------------------
---------------
Intercept                    -0.1936      1.155     -0.168      0.867      -2.
491       2.104
C(President)[T.RPresident]   -0.0320      0.028     -1.156      0.251      -0.
087       0.023
C(Senate)[T.RSenate]          0.0225      0.038      0.600      0.550      -0.
052       0.097
```

```
C(House)[T.RHouse]            -0.0873      0.042     -2.070      0.041      -0.
171      -0.003
Year                           0.0002      0.001      0.271      0.787      -0.
001       0.001
==============================================================================
Omnibus:                      110.302   Durbin-Watson:                   0.599
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2120.352
Skew:                           4.031   Prob(JB):                         0.00
Kurtosis:                      25.512   Cond. No.                     1.79e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
[2] The condition number is large, 1.79e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [11]:
```
def lin_reg_categorical_input(y, x_inputs):
    ols_string = f'Q("{y}") ~ '
    for x in x_inputs:
        ols_string += f'Q("{x}") +'
    ols_string = ols_string[:-1]
    model = ols(ols_string, data = df_data, missing = 'drop')
    fitted_model = model.fit()
    print(fitted_model.summary())
```

In [265]:
```
lin_reg_categorical_input("True Approval", ['President', 'House','Senate', 'Year'
lin_reg_categorical_input("Military Total", ['President', 'House','Senate']) #ok
lin_reg_categorical_input("Change in Military Total", ['President', 'House','Sena
lin_reg_categorical_input("Change in Adjusted Debt", ['President', 'House','Senat
lin_reg_categorical_input("Change in Adjusted Debt Percent", ['President', 'House
```

```
Q("Senate")[T.RSenate]           0.0210      0.037      0.568      0.571
-0.052       0.094
==============================================================================
=
Omnibus:                      108.899   Durbin-Watson:                   0.59
5
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2036.50
2
Skew:                           3.962   Prob(JB):                         0.0
0
Kurtosis:                      25.054   Cond. No.                         4.6
6
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [13]:
```
def lin_reg(y, x_factors):
    X = df_data[x_factors]
    Y = df_data[y]
    results = sm.OLS(Y,sm.add_constant(X), missing = 'drop').fit()
    print(results.summary())
```

In [14]:
```
lin_reg(["Change in Debt/GDP Ratio"],["Gov Marker"])
lin_reg(["True Approval"], ["Gov Marker", "Year"]) #indicative
lin_reg(["Military Total"],["Gov Marker"])
```

```
                                OLS Regression Results
==============================================================================
=====
Dep. Variable:     Change in Debt/GDP Ratio   R-squared:
0.019
Model:                                 OLS   Adj. R-squared:
0.007
Method:                      Least Squares   F-statistic:
1.645
Date:                     Sat, 16 Nov 2019   Prob (F-statistic):
0.203
Time:                             02:25:01   Log-Likelihood:                1
28.32
No. Observations:                      89   AIC:                           -
252.6
Df Residuals:                          87   BIC:                           -
247.7
Df Model:                               1
Covariance Type:                nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0024      0.009      0.280      0.780      -0.015       0.019
Gov Marker     0.0157      0.012      1.283      0.203      -0.009       0.040
==============================================================================
Omnibus:                       30.022   Durbin-Watson:                   0.854
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               92.214
Skew:                           1.053   Prob(JB):                     9.46e-21
Kurtosis:                       7.521   Cond. No.                         2.58
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:          True Approval   R-squared:                       0.214
Model:                            OLS   Adj. R-squared:                  0.174
Method:                 Least Squares   F-statistic:                     5.322
Date:                Sat, 16 Nov 2019   Prob (F-statistic):            0.00904
Time:                        02:25:01   Log-Likelihood:                -163.57
No. Observations:                  42   AIC:                             333.1
Df Residuals:                      39   BIC:                             338.4
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        929.5147    289.943      3.206      0.003     343.050    1515.980
Gov Marker    -3.4272      4.050     -0.846      0.403     -11.618       4.764
Year          -0.4475      0.145     -3.081      0.004      -0.741      -0.154
==============================================================================
Omnibus:                        3.091   Durbin-Watson:                   0.658
```

```
Prob(Omnibus):                  0.213   Jarque-Bera (JB):                 2.809
Skew:                           0.619   Prob(JB):                         0.245
Kurtosis:                       2.733   Cond. No.                      3.04e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
[2] The condition number is large, 3.04e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
                            OLS Regression Results
==============================================================================
Dep. Variable:         Military Total   R-squared:                       0.021
Model:                            OLS   Adj. R-squared:                  0.005
Method:                 Least Squares   F-statistic:                     1.298
Date:                Sat, 16 Nov 2019   Prob (F-statistic):              0.259
Time:                        02:25:01   Log-Likelihood:                -944.16
No. Observations:                  64   AIC:                             1892.
Df Residuals:                      62   BIC:                             1897.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.995e+06   9.68e+04     20.604      0.000     1.8e+06    2.19e+06
Gov Marker  1.882e+05   1.65e+05      1.139      0.259   -1.42e+05    5.18e+05
==============================================================================
Omnibus:                        5.193   Durbin-Watson:                   0.058
Prob(Omnibus):                  0.075   Jarque-Bera (JB):                3.939
Skew:                           0.479   Prob(JB):                        0.139
Kurtosis:                       2.252   Cond. No.                         2.42
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
```

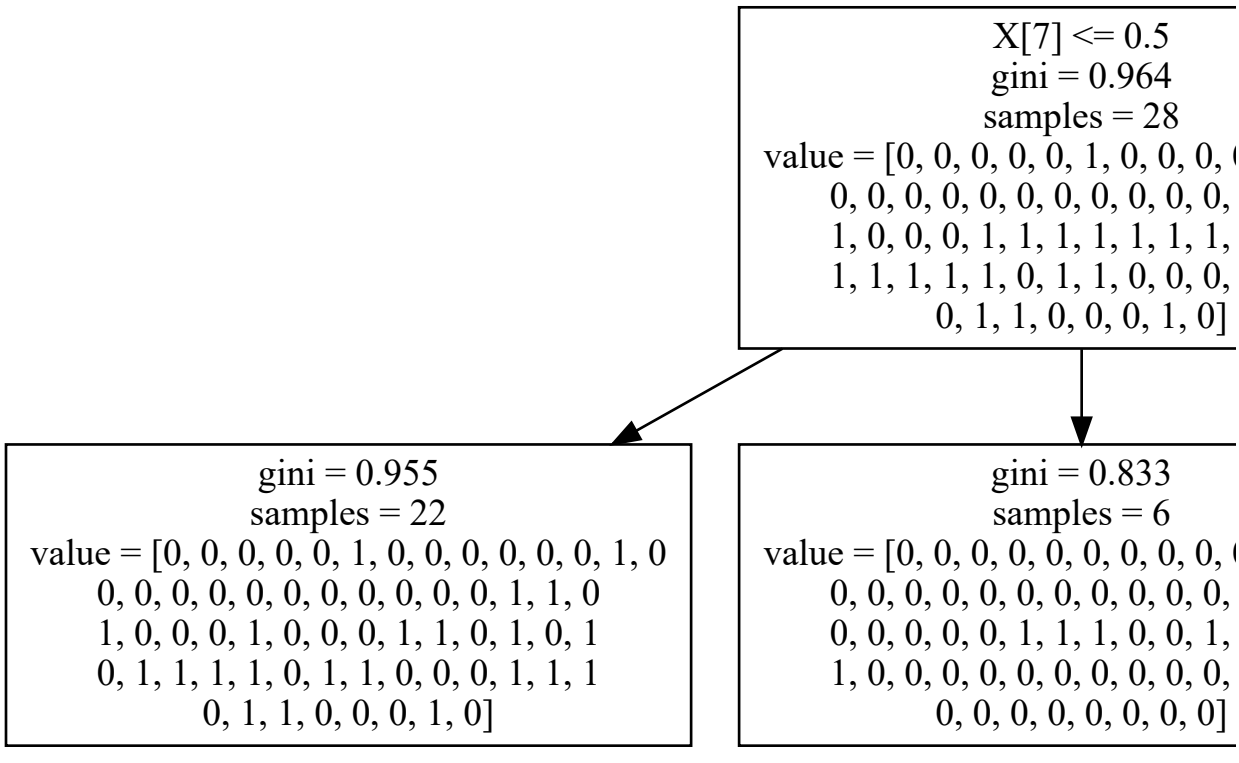
In [266]:
```
def cart_one_hot(y, x_input,depth = None):
    a = df_data[x_input+[y]].dropna()
    one_hot_data = pd.get_dummies(a[x_input], dummy_na=True)
    clf = tree.DecisionTreeClassifier(max_depth = depth)
    clf = clf.fit(one_hot_data, a[y])
    dot_data = tree.export_graphviz(clf, out_file=None)
    graph = graphviz.Source(dot_data)
    return(graph)
    # graph.render("iris")
#     tree.plot_tree(clf.fit(one_hot_data, a[y]))
#     plt.show()
```

In [267]:
```
df_data.columns
```

Out[267]:
```
Index(['Year', 'President', 'Senate', 'House', 'Divided/Unified', 'Gov Marker',
       'Party Marker', 'S&P Returns', 'True Approval', 'Debt',
       'Debt/GDP Ratio', 'Major Events by Presidential Term', 'Army', 'Navy',
       'Marine', 'Airforce', 'Military Total', 'Annual Average CPI Index',
       'Change in CPI Index Percent', 'Change in True Approval',
       'Change in Debt', 'Change in Debt/GDP Ratio', 'Change in Debt Percent',
       'Change in S&P Returns', 'Change in Military Total', 'Adjusted Debt',
       'Change in Adjusted Debt Percent', 'Change in Adjusted Debt'],
      dtype='object')
```

In [268]:
```
cart_one_hot("Military Total", ['President', 'House', 'Senate'])
```

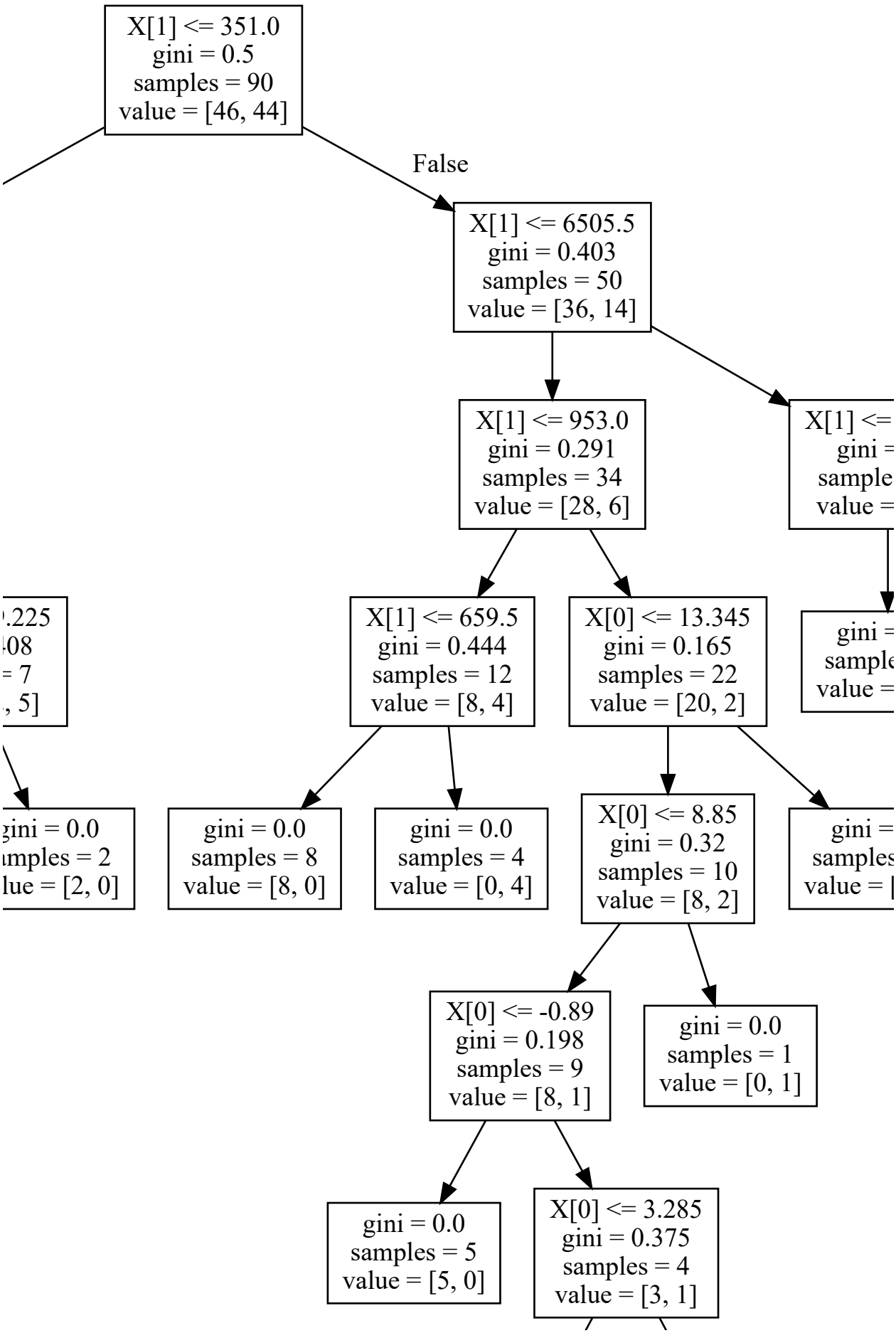
Out[268]:

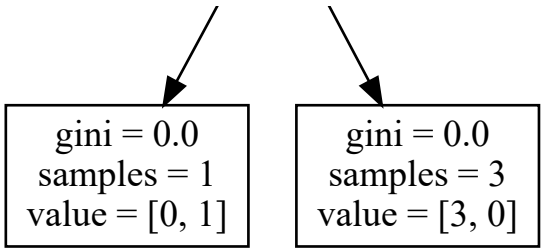


In [60]:
```
def cart(y, x_input, depth = None):
    a = df_data[x_input+[y]].dropna()
#     one_hot_data = pd.get_dummies(a[x_input], dummy_na=True)
    clf = tree.DecisionTreeClassifier(max_depth=depth)
    clf = clf.fit(a[x_input], a[y])
    dot_data = tree.export_graphviz(clf, out_file=None)
    graph = graphviz.Source(dot_data)
    return(graph)
```

In [61]:
```
cart("Divided/Unified", ['S&P Returns', 'Debt'])
```

Out[61]:





In [101]:
```
y = "Divided/Unified"
# x_input = ['Change in Military Total', 'Debt', 'True Approval'] #its ok
# x_input = ['Military Total', 'Change in Debt/GDP Ratio', 'True Approval'] #best
x_input = ['Military Total', 'Debt/GDP Ratio', 'True Approval']
df_data.columns
```

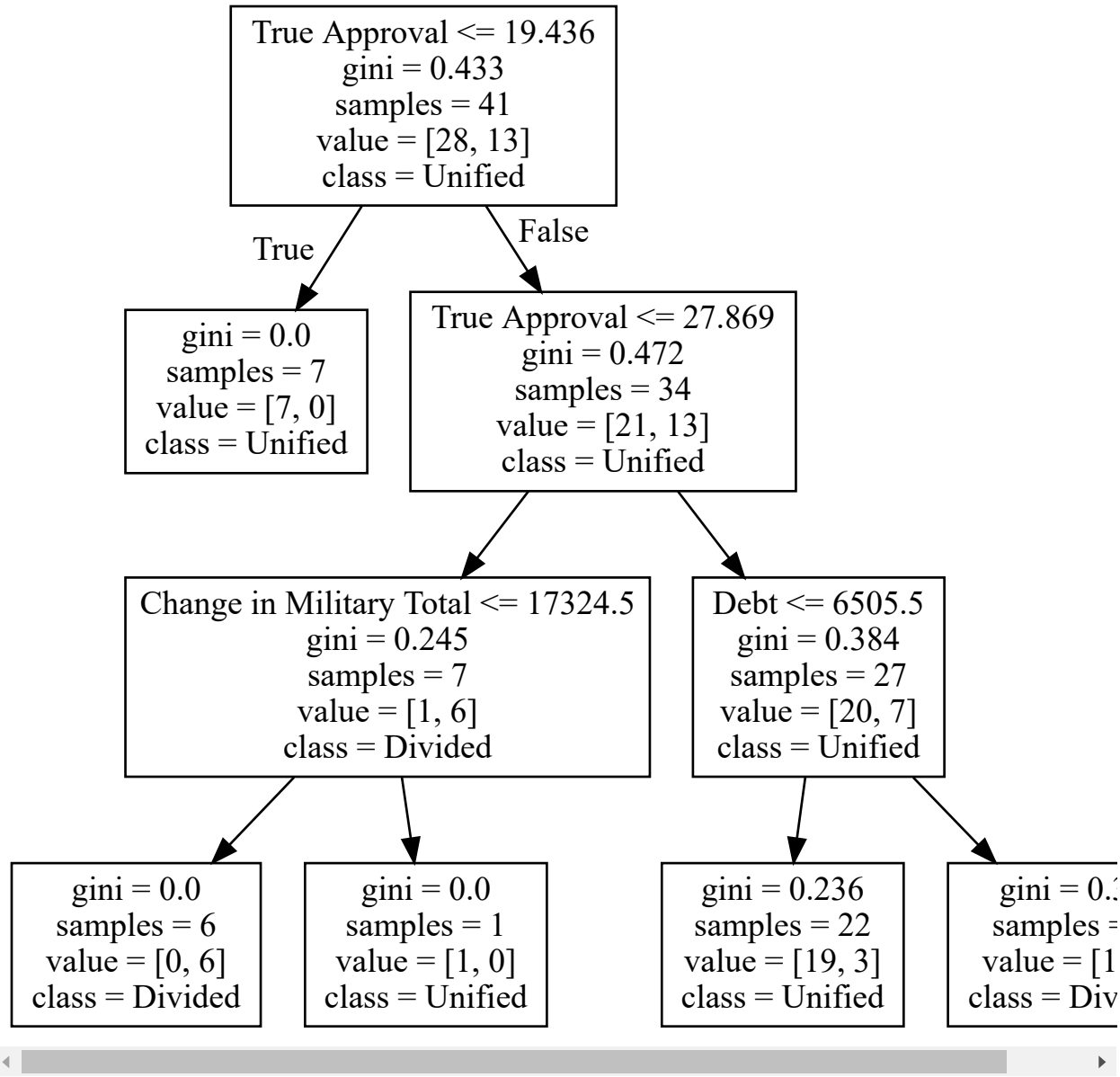
Out[101]:
```
Index(['Year', 'President', 'Senate', 'House', 'Divided/Unified', 'Gov Marker',
       'Party Marker', 'S&P Returns', 'True Approval', 'Debt',
       'Debt/GDP Ratio', 'Major Events by Presidential Term', 'Army', 'Navy',
       'Marine', 'Airforce', 'Military Total', 'Change in True Approval',
       'Change in Debt', 'Change in Debt/GDP Ratio', 'Change in Debt Percent',
       'Change in S&P Returns', 'Change in Military Total'],
      dtype='object')
```

In [177]:
```
def cart_validation_graph(y, x_input, depth = 10):
    parameters = {'max_depth':range(1,depth)}
    a = df_data[x_input+[y]].dropna()
    clf = GridSearchCV(tree.DecisionTreeClassifier(), parameters, n_jobs=4)
    clf = clf.fit(a[x_input], a[y])
    tree_model = clf.best_estimator_
    print (clf.best_score_, clf.best_params_)
    dot_data = tree.export_graphviz(tree_model, out_file=None, feature_names = x_
    graph = graphviz.Source(dot_data)
    return(graph)
```

In [110]:
```
cart_validation_graph("Divided/Unified", ['Change in Military Total', 'Debt', 'Tr
```

```
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)

0.6341463414634146 {'max_depth': 3}

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)
```

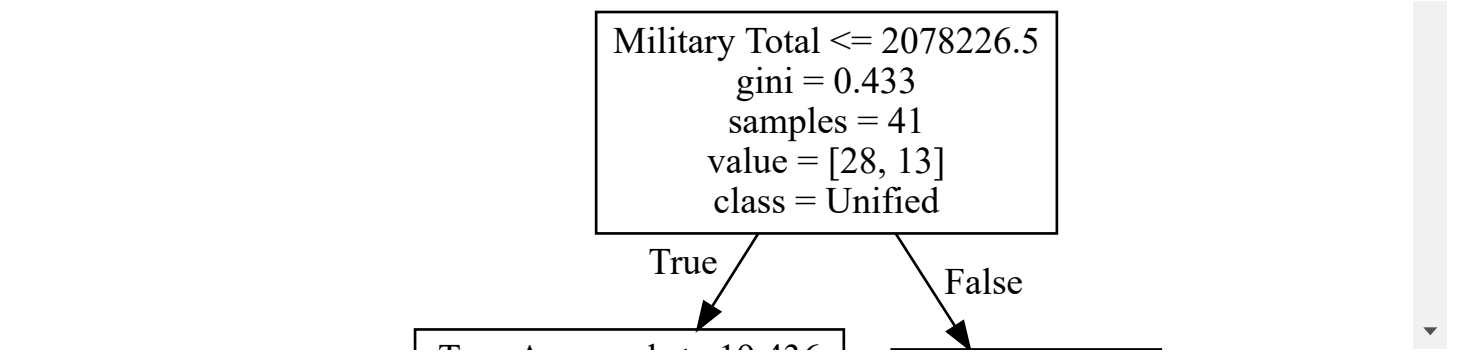
Out[110]:



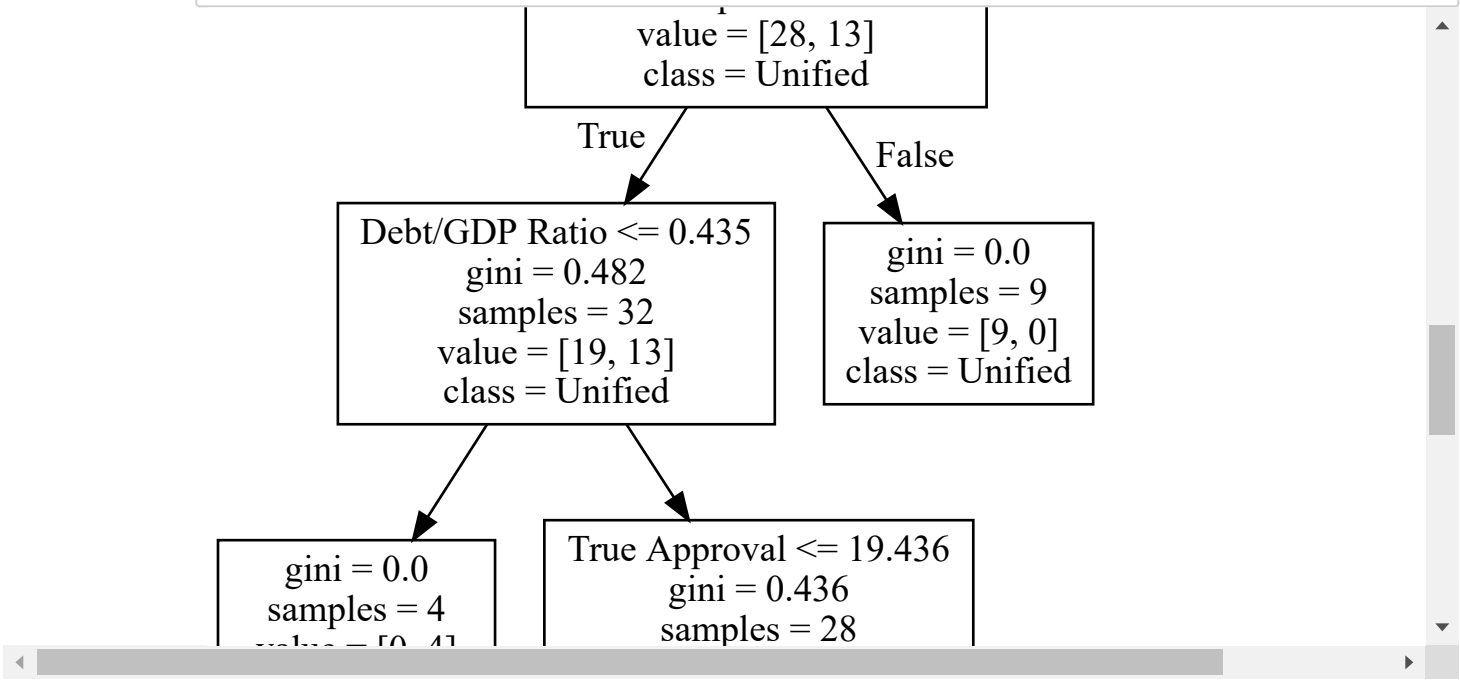
In [119]:
```
cart_validation_graph("Divided/Unified", ['Military Total', 'Change in Debt/GDP R
```

```
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)

0.7560975609756098 {'max_depth': 5}

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)
```

Out[119]:



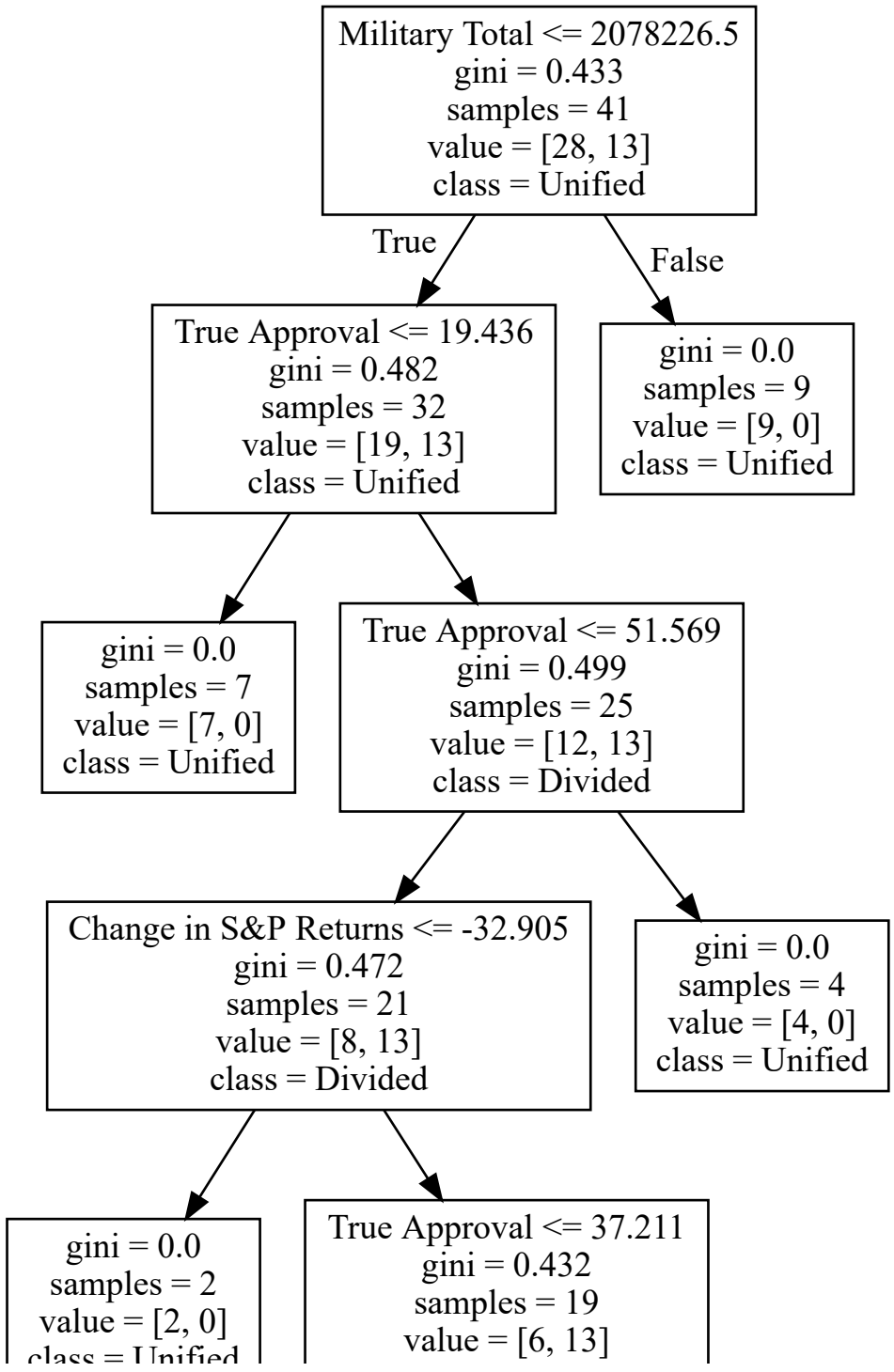
In [113]:
```
cart_validation_graph("Divided/Unified", ['Military Total', 'Debt/GDP Ratio', 'Tr
```

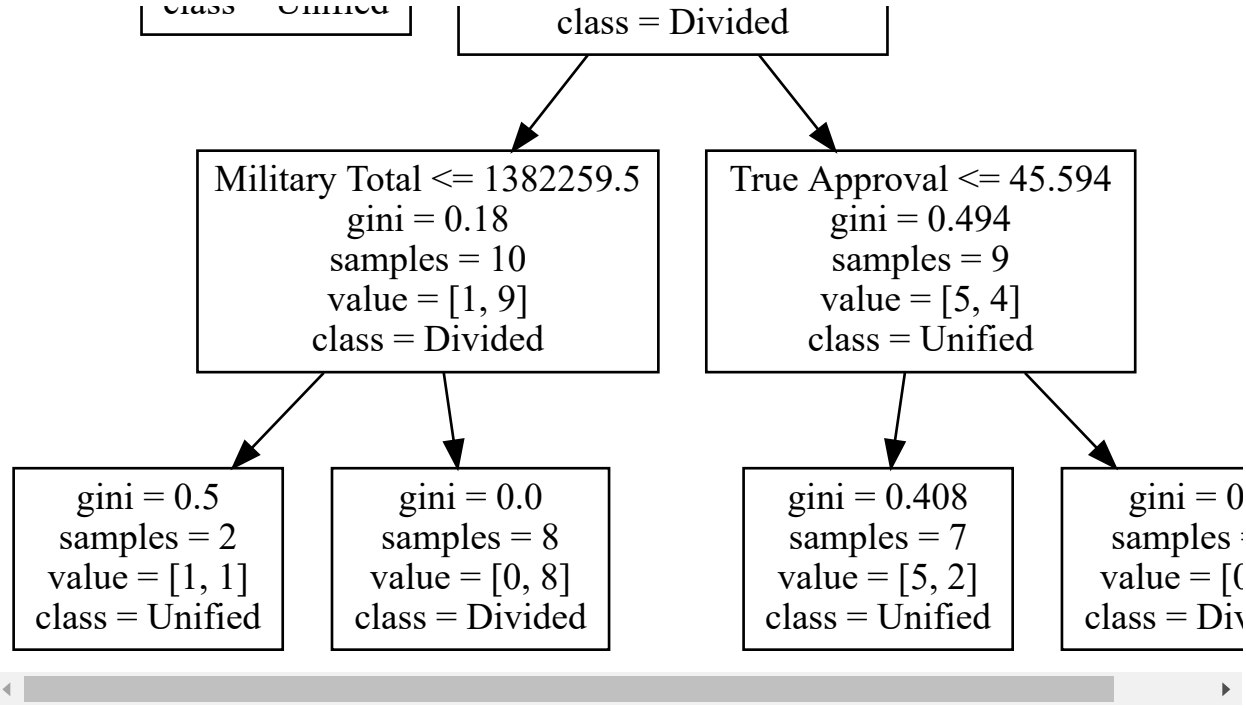


In [116]:
```
cart_validation_graph("Divided/Unified", ['Military Total',  'Change in S&P Retur
```

```
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)

0.7073170731707317 {'max_depth': 6}

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)
```

Out[116]:

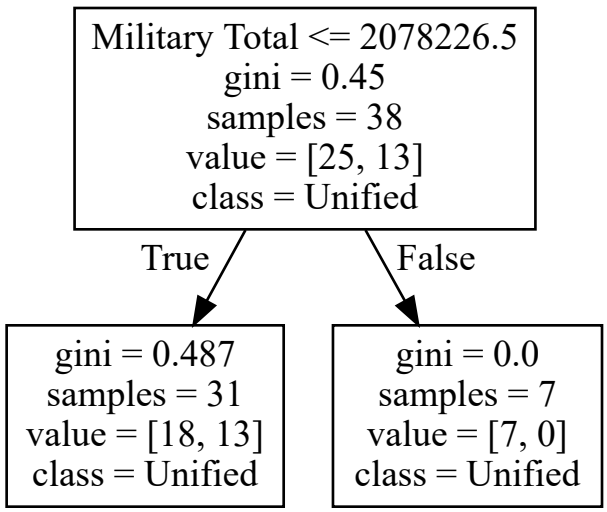




In [127]:
```
cart_validation_graph("Divided/Unified", ['Military Total',  'Change in S&P Retur
```

```
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)

0.6052631578947368 {'max_depth': 1}

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)
```

Out[127]:



In [296]:
```
list(df_data.columns)
```

Out[296]:
```
['Year',
 'President',
 'Senate',
 'House',
 'Divided/Unified',
 'Gov Marker',
 'Party Marker',
 'S&P Returns',
 'True Approval',
 'Debt',
 'Debt/GDP Ratio',
 'Major Events by\xa0Presidential Term',
 'Army',
 'Navy',
 'Marine',
 'Airforce',
 'Military Total',
 'Annual Average CPI Index',
 'Change in CPI Index Percent',
 'GDP Growth',
 'Real GDP',
 'Change in True Approval',
 'Change in Debt',
 'Change in Debt/GDP Ratio',
 'Change in Debt Percent',
 'Change in S&P Returns',
 'Change in Military Total',
 'Adjusted Debt',
 'Change in Adjusted Debt Percent',
 'Change in Adjusted Debt',
 'Change in Real GDP']
```

In [299]:
```
import itertools
relevant_data = list(df_data.columns)[16:] + list(df_data.columns)[7:][:4]
# print(list(df_data.columns)[16:])
# print(list(df_data.columns)[7:][:4])

def valid_combos(relevant_data, size):
    subsets_tuples = list(itertools.combinations(relevant_data, size))
    subsets_tuples_copy = subsets_tuples.copy()
    for subset in subsets_tuples_copy:
        for element in subset:
            subset_copy_remove = list(subset).copy()
            subset_copy_remove.remove(element)
#             print(element, subset)
#             print(list(subset).remove(element))
            if any(element in s for s in subset_copy_remove): #dont put delta and
                subsets_tuples.remove(subset)
                break
    return subsets_tuples

valid_x_inputs = valid_combos(relevant_data, 3)
```

In [300]:
```
def cart_validation(y, x_input, depth = 7):
    parameters = {'max_depth':range(1,depth)}
    a = df_data[x_input+[y]].dropna()
    clf = GridSearchCV(tree.DecisionTreeClassifier(), parameters, n_jobs=4)
    clf = clf.fit(a[x_input], a[y])
    tree_model = clf.best_estimator_
    return((clf.best_score_, clf.best_params_['max_depth']))
```

In [301]:
```
from tqdm import tqdm_notebook as tqdm

cart_results_dict = {}
for inputs in tqdm(valid_x_inputs, total = len(valid_x_inputs)):
    results = cart_validation("Divided/Unified", list(inputs))
    if results[0] > 0.5:
        cart_results_dict[inputs] = results
```

```
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
1978: FutureWarning: The default value of cv will change from 3 to 5 in versi
on 0.22. Specify it explicitly to silence this warning.
```

In [302]:
```
sorted_results_dict = sorted(cart_results_dict.items(), key=lambda x: x[1], rever
print(sorted_results_dict[:30])
```

```
952380952381, 5)), (('Change in Military Total', 'True Approval', 'Debt/GDP R
atio'), (0.7317073170731707, 4)), (('Change in Debt/GDP Ratio', 'Change in S&
P Returns', 'True Approval'), (0.7142857142857143, 5)), (('Change in S&P Retu
rns', 'True Approval', 'Debt/GDP Ratio'), (0.7142857142857143, 5)), (('Change
in Adjusted Debt', 'True Approval', 'Debt/GDP Ratio'), (0.7142857142857143,
5)), (('Change in CPI Index Percent', 'Change in Real GDP', 'True Approval'),
(0.7142857142857143, 4)), (('Change in Real GDP', 'S&P Returns', 'True Approv
al'), (0.7142857142857143, 4)), (('Military Total', 'True Approval', 'Debt/GD
P Ratio'), (0.7073170731707317, 5)), (('Military Total', 'Change in S&P Retur
ns', 'True Approval'), (0.7073170731707317, 4)), (('Change in S&P Returns',
'Change in Military Total', 'True Approval'), (0.7073170731707317, 4)), (('Ch
ange in Debt/GDP Ratio', 'Change in Military Total', 'True Approval'), (0.707
3170731707317, 2)), (('Change in Military Total', 'Change in Adjusted Debt',
'True Approval'), (0.7073170731707317, 2)), (('Change in CPI Index Percent',
'Change in Debt/GDP Ratio', 'True Approval'), (0.6904761904761905, 6)), (('Ch
ange in Debt/GDP Ratio', 'Change in Adjusted Debt', 'True Approval'), (0.6904
761904761905, 6)), (('Change in S&P Returns', 'True Approval', 'Debt'), (0.69
04761904761905, 6)), (('Change in Debt Percent', 'Adjusted Debt', 'True Appro
val'), (0.6904761904761905, 5)), (('S&P Returns', 'True Approval', 'Debt/GDP
```

In [225]:
```
def cart_validation_graph_export(y, x_input, i, depth = 10):
    parameters = {'max_depth':range(1,depth)}
    a = df_data[x_input+[y]].dropna()
    clf = GridSearchCV(tree.DecisionTreeClassifier(), parameters, n_jobs=4)
    clf = clf.fit(a[x_input], a[y])
    tree_model = clf.best_estimator_
    print (clf.best_score_, clf.best_params_)
    dot_data = tree.export_graphviz(tree_model, out_file=None, feature_names = x_
    graph = graphviz.Source(dot_data)
    graph.render(f'{", ".join(x_input).replace("/","-")}')
```

In [279]:
```
for i in range(10):
    x_input = list(sorted_results_dict[i][0])
    print(x_input)
    y = "Divided/Unified"
    cart_validation_graph_export(y, x_input,i, depth = 6)
```

```
['Change in Debt/GDP Ratio', 'Change in Military Total', 'True Approval']

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)

0.6585365853658537 {'max_depth': 2}
['Military Total', 'Change in Debt/GDP Ratio', 'True Approval']

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)

0.7073170731707317 {'max_depth': 5}
['Change in Military Total', 'True Approval', 'Debt/GDP Ratio']

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)

0.7560975609756098 {'max_depth': 4}
['Change in Debt/GDP Ratio', 'Change in S&P Returns', 'True Approval']

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)

0.7142857142857143 {'max_depth': 5}
```

```
['Military Total', 'Change in S&P Returns', 'True Approval']
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)

0.7073170731707317 {'max_depth': 5}
['Change in Military Total', 'S&P Returns', 'True Approval']

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)

0.7317073170731707 {'max_depth': 5}
['Change in Debt/GDP Ratio', 'Change in Debt Percent', 'Change in Military Tota
l']

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)

0.6984126984126984 {'max_depth': 4}
['Change in CPI Index Percent', 'True Approval', 'Debt/GDP Ratio']

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)

0.6666666666666666 {'max_depth': 2}
['Change in S&P Returns', 'True Approval', 'Debt']

C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)
```

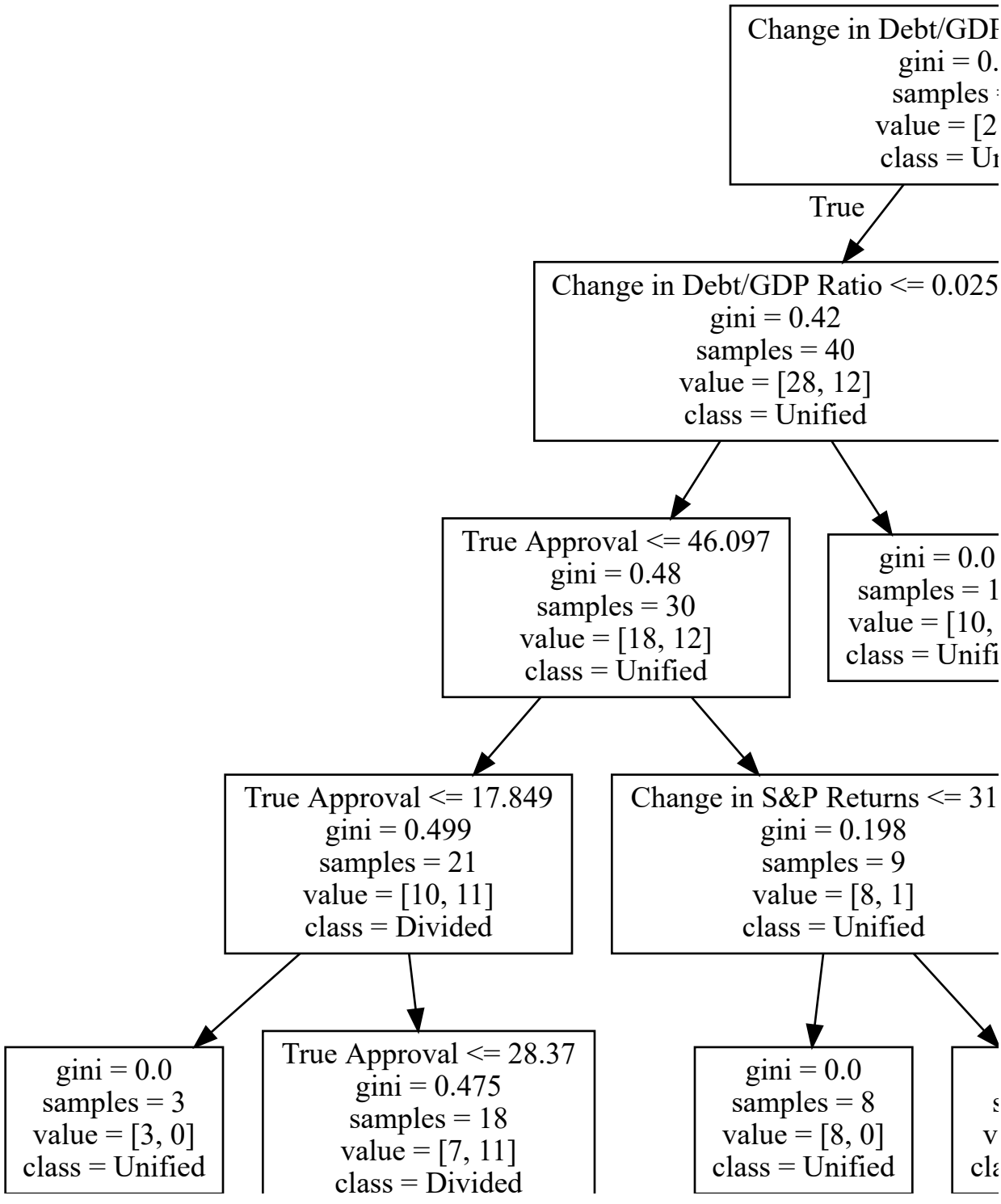
```
0.6904761904761905 {'max_depth': 5}
['Change in CPI Index Percent', 'Change in Debt/GDP Ratio', 'True Approval']
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:1978: FutureWarning: The default value of cv will change from 3 to 5 in version 0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:814: DeprecationWarning: The default of the `iid` parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change numeric results when test-set sizes are unequal.
  DeprecationWarning)
```

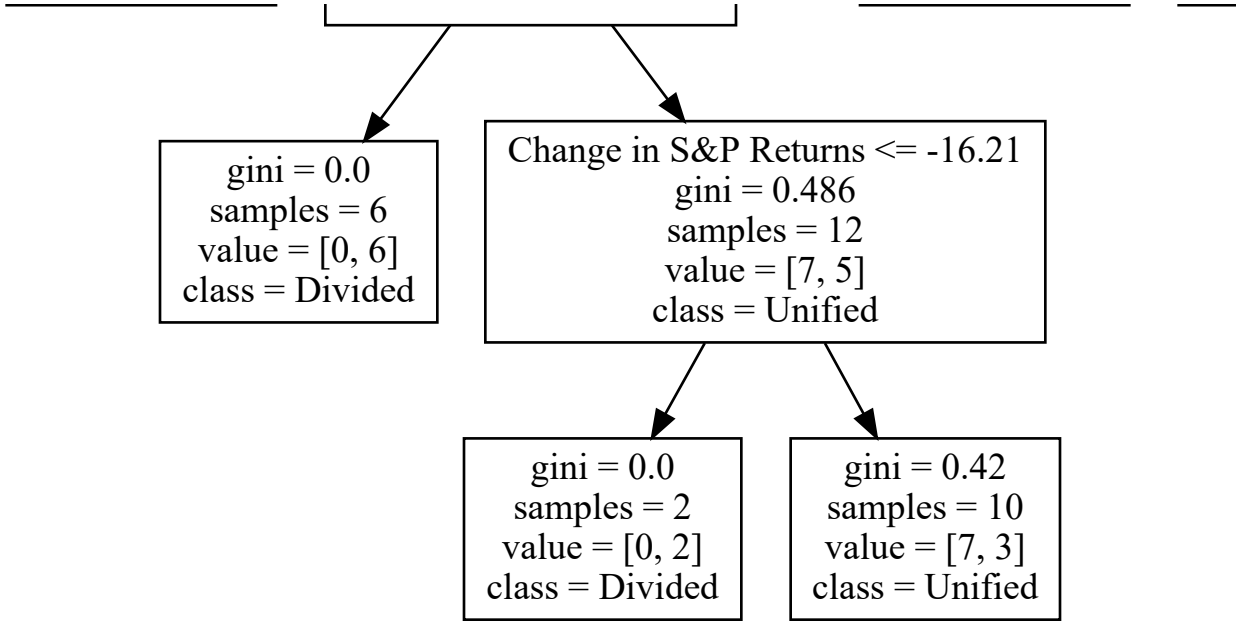
```
0.6904761904761905 {'max_depth': 2}
```

In [182]:
```
cart_validation_graph("Divided/Unified", ['Change in Debt/GDP Ratio', 'Change in
```

```
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:19
78: FutureWarning: The default value of cv will change from 3 to 5 in version
0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
C:\Users\shawn\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:8
14: DeprecationWarning: The default of the `iid` parameter will change from Tru
e to False in version 0.22 and will be removed in 0.24. This will change numeri
c results when test-set sizes are unequal.
  DeprecationWarning)

0.7142857142857143 {'max_depth': 6}
```

Out[182]:





In [209]:
```
>>> from sklearn.datasets import load_iris
>>> from sklearn import tree
>>> iris = load_iris()
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(iris.data, iris.target)
>>> dot_data = tree.export_graphviz(clf, out_file=None)
>>> graph = graphviz.Source(dot_data)
>>> graph.render("iris")
```

Out[209]:
```
'iris.pdf'
```

In [130]:
```
from sklearn.datasets import load_iris
>>> from sklearn.tree import DecisionTreeClassifier
>>> from sklearn.tree.export import export_text
>>> iris = load_iris()
>>> X = iris['data']
>>> y = iris['target']
>>> decision_tree = DecisionTreeClassifier(random_state=0, max_depth=2)
>>> decision_tree = decision_tree.fit(X, y)
>>> r = export_text(decision_tree, feature_names=iris['feature_names'])
>>> print(r)
```

```
|--- petal width (cm) <= 0.80
|   |--- class: 0
|--- petal width (cm) >  0.80
|   |--- petal width (cm) <= 1.75
|   |   |--- class: 1
|   |--- petal width (cm) >  1.75
|   |   |--- class: 2
```

In [35]:
```
print(iris['feature_names'])
iris.target_names
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (c
m)']
```

Out[35]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [29]:
```
>>> from sklearn.datasets import load_iris
>>> from sklearn import tree
>>> iris = load_iris()
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(iris.data, iris.target)
iris
```

```
        [6.4, 3.1, 5.5, 1.8],
        [6. , 3. , 4.8, 1.8],
        [6.9, 3.1, 5.4, 2.1],
        [6.7, 3.1, 5.6, 2.4],
        [6.9, 3.1, 5.1, 2.3],
        [5.8, 2.7, 5.1, 1.9],
        [6.8, 3.2, 5.9, 2.3],
        [6.7, 3.3, 5.7, 2.5],
        [6.7, 3. , 5.2, 2.3],
        [6.3, 2.5, 5. , 1.9],
        [6.5, 3. , 5.2, 2. ],
        [6.2, 3.4, 5.4, 2.3],
        [5.9, 3. , 5.1, 1.8]]),
 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

        1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
```

In [ ]:
```
```

In [27]:
```
>>> dot_data = tree.export_graphviz(clf, out_file=None,
...                                 feature_names=iris.feature_names,
...                                 class_names=iris.target_names,
...                                 filled=True, rounded=True,
...                                 special_characters=True)
>>> graph = graphviz.Source(dot_data)
>>> graph
```

Out[27]:

petal wi
gin
sam

In [ ]: