Rahul Sharma

Predicition Using Decision Tree Algorithm (Iris dataset)

In [89]:

```
import pandas as pd
import numpy as np
%matplotlib inline
```

In [90]:

```
df= pd.read_csv(r"C:\Users\Rahul\Desktop\iris.csv")
df.head()
```

Out[90]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [91]:

```
df.count()
```

Out[91]:

```
sepal_length    150
sepal_width     150
petal_length    150
petal_width     150
species         150
dtype: int64
```

In [92]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [93]:

```
df.dtypes
```

Out[93]:

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

In [94]:

```
df.shape
```

Out[94]:

```
(150, 5)
```

In [95]:

```
df.groupby(['sepal_length','sepal_width']).size()
```

Out[95]:

```
sepal_length  sepal_width
4.3           3.0            1
4.4           2.9            1
              3.0            1
              3.2            1
4.5           2.3            1
                            ..
7.7           2.6            1
              2.8            1
              3.0            1
              3.8            1
7.9           3.8            1
Length: 116, dtype: int64
```

In [96]:

```
df.groupby(['petal_length','petal_width']).size()
```

Out[96]:

```
petal_length  petal_width
1.0           0.2            1
1.1           0.1            1
1.2           0.2            2
1.3           0.2            4
              0.3            2
                            ..
6.4           2.0            1
6.6           2.1            1
6.7           2.0            1
              2.2            1
6.9           2.3            1
Length: 102, dtype: int64
```

In [97]:

```
df.groupby('species').size()
```

Out[97]:

```
species
setosa        50
versicolor    50
virginica     50
dtype: int64
```

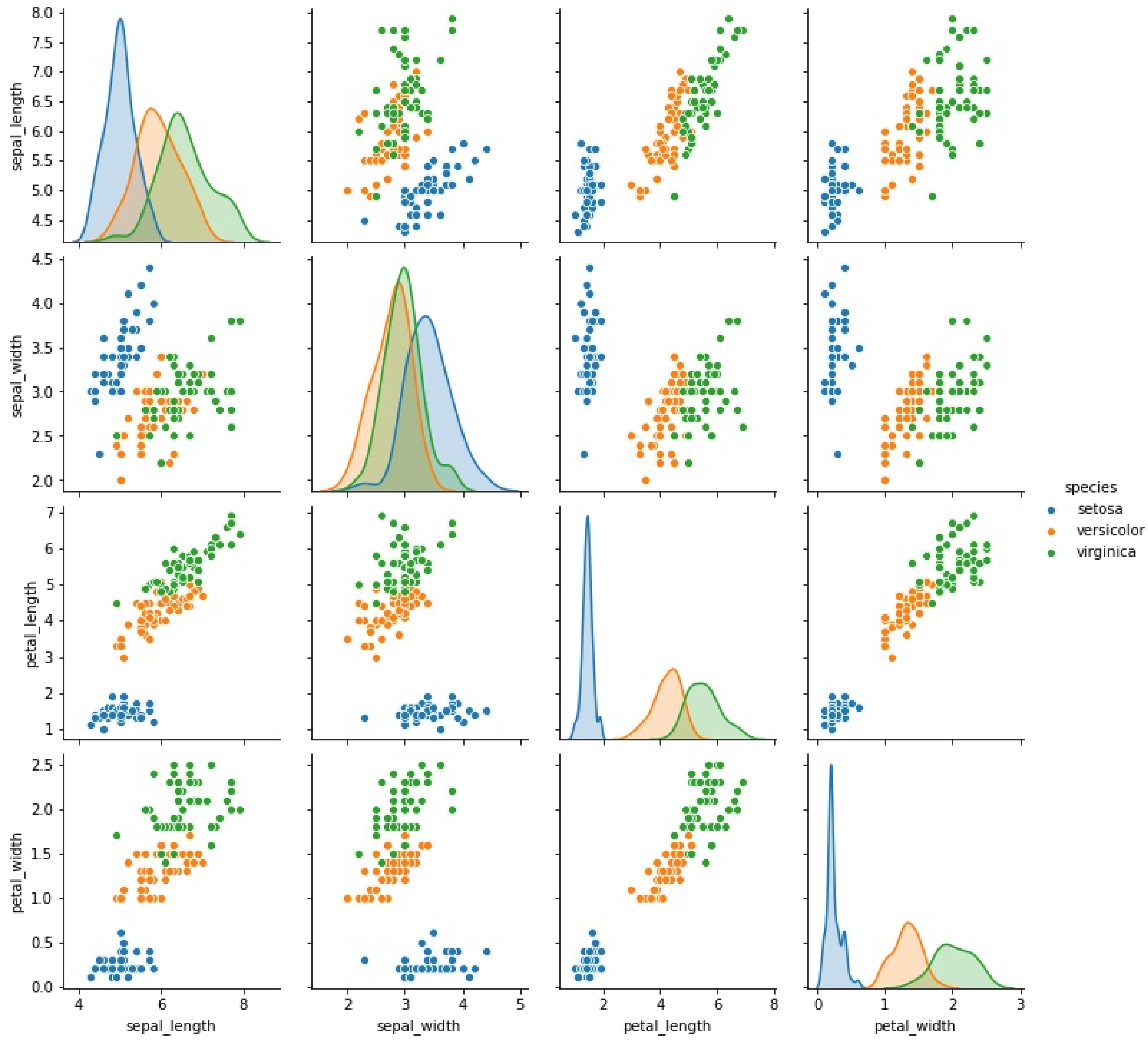
Pair Plotting dependent Variables

In [98]:

```
import seaborn as sns
sns.pairplot(df,hue='species')
```

Out[98]:

```
<seaborn.axisgrid.PairGrid at 0x378b4b4820>
```



Label Encoding for Dependent variables

In [99]:

```
from sklearn import preprocessing as prep
label_encoder = prep.LabelEncoder()
df['species'] = label_encoder.fit_transform(df['species'])
df['species'].unique()
```

Out[99]:

```
array([0, 1, 2])
```

In [100]:

```
df.head()
```

Out[100]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Declaring Independent and dependent parameters under variables

In [101]:

```
x=df[['sepal_length','sepal_width','petal_length','petal_width']]
y=df[['species']]
```

Creating and Splitting train and test data

In [102]:

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.1,random_state=0)
```

Decision Tree Classification Model

In [111]:

```
from sklearn.tree import DecisionTreeClassifier as dt
model=dt(max_depth=10,random_state=100)
model.fit(x_train,y_train)
```

Out[111]:

```
DecisionTreeClassifier(max_depth=10, random_state=100)
```

Feature Scaling

In [112]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train=sc.fit_transform(x_train)
x_test =sc.transform(x_test)
print(x_train)
```

```
[[-1.08593443  0.09813652 -1.24448561 -1.41148027]
 [ 0.2219428  -0.36672067  0.44627813  0.40699124]
 [-0.37254685 -1.06400644  0.38991934  0.01731878]
 [-1.20483236 -0.13429207 -1.3008444  -1.15169862]
 [-0.49144478  1.95756524 -1.35720319 -1.0218078 ]
 [-0.25364892 -0.59914926  0.6717133   1.05644535]
 [-0.25364892 -0.13429207  0.44627813  0.40699124]
 [-1.20483236  0.79542229 -1.01905044 -1.28158945]
 [-1.68042408 -0.36672067 -1.3008444  -1.28158945]
 [ 0.45973866 -0.59914926  0.61535451  0.79666371]
 [-1.44262822  1.26027947 -1.52627956 -1.28158945]
 [-0.84813857  1.72513665 -1.01905044 -1.0218078 ]
 [ 0.45973866 -0.36672067  0.33356055  0.1472096 ]
 [-0.9670365  -1.76129221 -0.23002736 -0.24246287]
 [-0.9670365   0.79542229 -1.18812682 -1.0218078 ]
 [ 0.69753452  0.09813652  1.00986605  0.79666371]
 [-0.49144478 -0.13429207  0.44627813  0.40699124]
 [-0.72924064  1.02785088 -1.24448561 -1.28158945]
 [ 0.34084073 -0.13429207  0.6717133   0.79666371]
 [ 0.69753452 -0.59914926  1.06622484  1.316227  ]
 [-0.72924064 -0.83157785  0.10812538  0.27710042]
 [-0.13475099  1.72513665 -1.13176803 -1.15169862]
 [ 0.2219428  -0.83157785  0.78443088  0.53688206]
 [ 0.10304487 -0.13429207  0.27720176  0.40699124]
 [-0.01585306 -1.06400644  0.16448418  0.01731878]
 [ 1.17312624 -0.13429207  1.00986605  1.18633618]
 [-1.32373029  0.33056511 -1.35720319 -1.28158945]
 [ 1.29202417  0.09813652  0.78443088  1.44611782]
 [-0.9670365   1.02785088 -1.18812682 -0.76202616]
 [-0.49144478  1.49270806 -1.24448561 -1.28158945]
 [-0.9670365  -2.45857798 -0.11730978 -0.24246287]
 [ 0.81643245 -0.13429207  1.00986605  0.79666371]
 [ 1.05422831  0.5629937   1.12258363  1.70589946]
 [ 0.2219428  -1.9937208   0.72807209  0.40699124]
 [ 1.05422831 -1.29643503  1.17894242  0.79666371]
 [-0.25364892 -1.29643503  0.10812538 -0.11257205]
 [ 2.24320761 -0.13429207  1.34801879  1.44611782]
 [ 0.57863659  0.5629937   0.55899572  0.53688206]
 [-0.37254685 -1.52886362  0.05176659 -0.11257205]
```

```
 [ 0.57863659 -0.83157785  0.6717133   0.79666371]
 [ 0.57863659 -0.59914926  0.78443088  0.40699124]
 [-1.08593443 -1.29643503  0.44627813  0.66677289]
 [ 0.57863659 -1.29643503  0.72807209  0.92655453]
 [ 1.4109221   0.33056511  0.55899572  0.27710042]
 [ 0.81643245 -0.13429207  0.84078967  1.05644535]
 [ 0.2219428   0.79542229  0.44627813  0.53688206]
 [-1.20483236  0.09813652 -1.18812682 -1.28158945]
 [-0.01585306 -0.83157785  0.78443088  0.92655453]
 [-0.25364892 -0.83157785  0.27720176  0.1472096 ]
 [-0.25364892 -0.36672067 -0.06095099  0.1472096 ]
 [-0.37254685 -1.29643503  0.16448418  0.1472096 ]
 [ 0.34084073 -0.13429207  0.50263692  0.27710042]
 [ 1.64871796  0.33056511  1.29166     0.79666371]
 [-0.61034271  1.49270806 -1.24448561 -1.28158945]
 [-1.79932201 -0.13429207 -1.46992077 -1.41148027]
 [ 0.69753452 -0.83157785  0.89714846  0.92655453]
 [-0.13475099 -0.13429207  0.27720176  0.01731878]
 [-0.49144478  0.79542229 -1.13176803 -1.28158945]
 [-0.13475099  3.1197082  -1.24448561 -1.0218078 ]
 [ 1.29202417  0.09813652  0.6717133   0.40699124]
 [-1.44262822  0.09813652 -1.24448561 -1.28158945]
 [ 0.10304487 -0.13429207  0.78443088  0.79666371]
 [-0.84813857 -1.29643503 -0.39910374 -0.11257205]
 [-1.44262822  0.79542229 -1.3008444  -1.15169862]
 [ 0.45973866 -1.9937208   0.44627813  0.40699124]
 [ 1.64871796  1.26027947  1.34801879  1.70589946]
 [-0.13475099 -0.36672067  0.27720176  0.1472096 ]
 [-1.20483236 -0.13429207 -1.3008444  -1.41148027]
 [ 1.52982003 -0.13429207  1.23530121  1.18633618]
 [ 1.29202417  0.33056511  1.12258363  1.44611782]
 [ 0.81643245 -0.13429207  1.17894242  1.316227  ]
 [ 0.69753452 -0.59914926  1.06622484  1.18633618]
 [-0.84813857  1.72513665 -1.18812682 -1.28158945]
 [-1.20483236  0.79542229 -1.18812682 -1.28158945]
 [ 0.81643245  0.33056511  0.78443088  1.05644535]
 [ 1.05422831  0.5629937   1.12258363  1.18633618]
 [-1.56152615 -1.76129221 -1.35720319 -1.15169862]
 [ 0.45973866  0.79542229  0.95350726  1.44611782]
 [-1.08593443 -0.13429207 -1.3008444  -1.28158945]
 [-0.13475099 -1.29643503  0.72807209  1.05644535]
 [ 1.29202417  0.09813652  0.95350726  1.18633618]
 [-1.68042408  0.33056511 -1.35720319 -1.28158945]
 [-0.9670365   1.26027947 -1.3008444  -1.28158945]
 [ 1.64871796 -0.13429207  1.17894242  0.53688206]
 [-0.84813857  1.02785088 -1.3008444  -1.15169862]
 [-1.68042408 -0.13429207 -1.35720319 -1.28158945]
 [-0.49144478  1.95756524 -1.13176803 -1.0218078 ]
 [-0.37254685 -1.76129221  0.16448418  0.1472096 ]
 [ 1.17312624  0.33056511  1.23530121  1.44611782]
 [ 2.12430968 -0.13429207  1.62981275  1.18633618]
 [-0.84813857  1.02785088 -1.3008444  -1.28158945]
 [-1.08593443  0.09813652 -1.24448561 -1.41148027]
 [-0.72924064  0.79542229 -1.3008444  -1.28158945]
 [-0.13475099 -0.59914926  0.44627813  0.1472096 ]
 [ 0.93533038 -0.13429207  0.38991934  0.27710042]
 [-0.9670365   0.33056511 -1.41356198 -1.28158945]
 [-0.84813857  0.5629937  -1.13176803 -0.89191698]
 [ 0.69753452 -0.36672067  0.33356055  0.1472096 ]
 [-0.49144478  0.79542229 -1.24448561 -1.0218078 ]
 [ 2.24320761 -1.06400644  1.79888913  1.44611782]
 [-1.08593443 -1.52886362 -0.23002736 -0.24246287]
 [ 2.48100347  1.72513665  1.51709517  1.05644535]
 [ 1.05422831  0.09813652  0.38991934  0.27710042]
 [-0.72924064  2.42242242 -1.24448561 -1.41148027]
 [ 0.2219428  -0.13429207  0.61535451  0.79666371]
 [-0.01585306  2.18999383 -1.41356198 -1.28158945]
 [ 2.24320761 -0.59914926  1.68617154  1.05644535]
 [-0.84813857  1.72513665 -1.24448561 -1.15169862]
 [-1.32373029  0.33056511 -1.18812682 -1.28158945]
 [ 1.88651382 -0.59914926  1.34801879  0.92655453]
 [-0.9670365   0.5629937  -1.3008444  -1.28158945]
 [ 0.57863659  0.79542229  1.06622484  1.57600864]
 [-0.13475099 -0.59914926  0.22084297  0.1472096 ]
 [-0.01585306 -0.83157785  0.10812538  0.01731878]
 [-0.13475099 -1.06400644 -0.11730978 -0.24246287]
 [ 0.69753452  0.33056511  0.89714846  1.44611782]
 [ 1.05422831 -0.13429207  0.84078967  1.44611782]
 [ 0.57863659 -1.29643503  0.6717133   0.40699124]
 [ 1.05422831 -0.13429207  0.72807209  0.66677289]
 [-0.9670365  -0.13429207 -1.18812682 -1.28158945]
 [-0.37254685 -1.52886362 -0.0045922  -0.24246287]
 [ 1.05422831  0.09813652  1.06622484  1.57600864]
```

```
 [-0.01585306 -0.83157785  0.78443088  0.92655453]
 [-0.84813857  0.79542229 -1.24448561 -1.28158945]
 [ 0.93533038 -0.36672067  0.50263692  0.1472096 ]
 [-0.25364892 -0.13429207  0.22084297  0.1472096 ]
 [ 0.10304487  0.33056511  0.61535451  0.79666371]
 [ 0.57863659 -1.76129221  0.38991934  0.1472096 ]
 [-0.37254685  1.02785088 -1.35720319 -1.28158945]
 [-0.84813857  1.49270806 -1.24448561 -1.0218078 ]
 [-1.08593443  0.09813652 -1.24448561 -1.41148027]
 [ 0.57863659 -0.36672067  1.06622484  0.79666371]
 [-0.01585306 -0.83157785  0.22084297 -0.24246287]
 [ 2.24320761  1.72513665  1.68617154  1.316227  ]
 [-1.44262822  0.33056511 -1.3008444  -1.28158945]]
```

Predicting Model

In [113]:

```
predictions=model.predict(x_test)
predictions
```

Out[113]:

```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1])
```

Classificatin report, Confusion Matrix

In [114]:

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,predictions))
```

```
[[3 0 0]
 [0 8 0]
 [0 0 4]]
```

In [115]:

```
print(confusion_matrix(y_test,predictions))
```

```
[[3 0 0]
 [0 8 0]
 [0 0 4]]
```

Predicting the Accuracy of the model

In [116]:

```
from sklearn import metrics
print('The accuracy of the DecisionTreeClassifier is:',metrics.accuracy_score(y_test,predictions))
```

```
The accuracy of the DecisionTreeClassifier is: 1.0
```

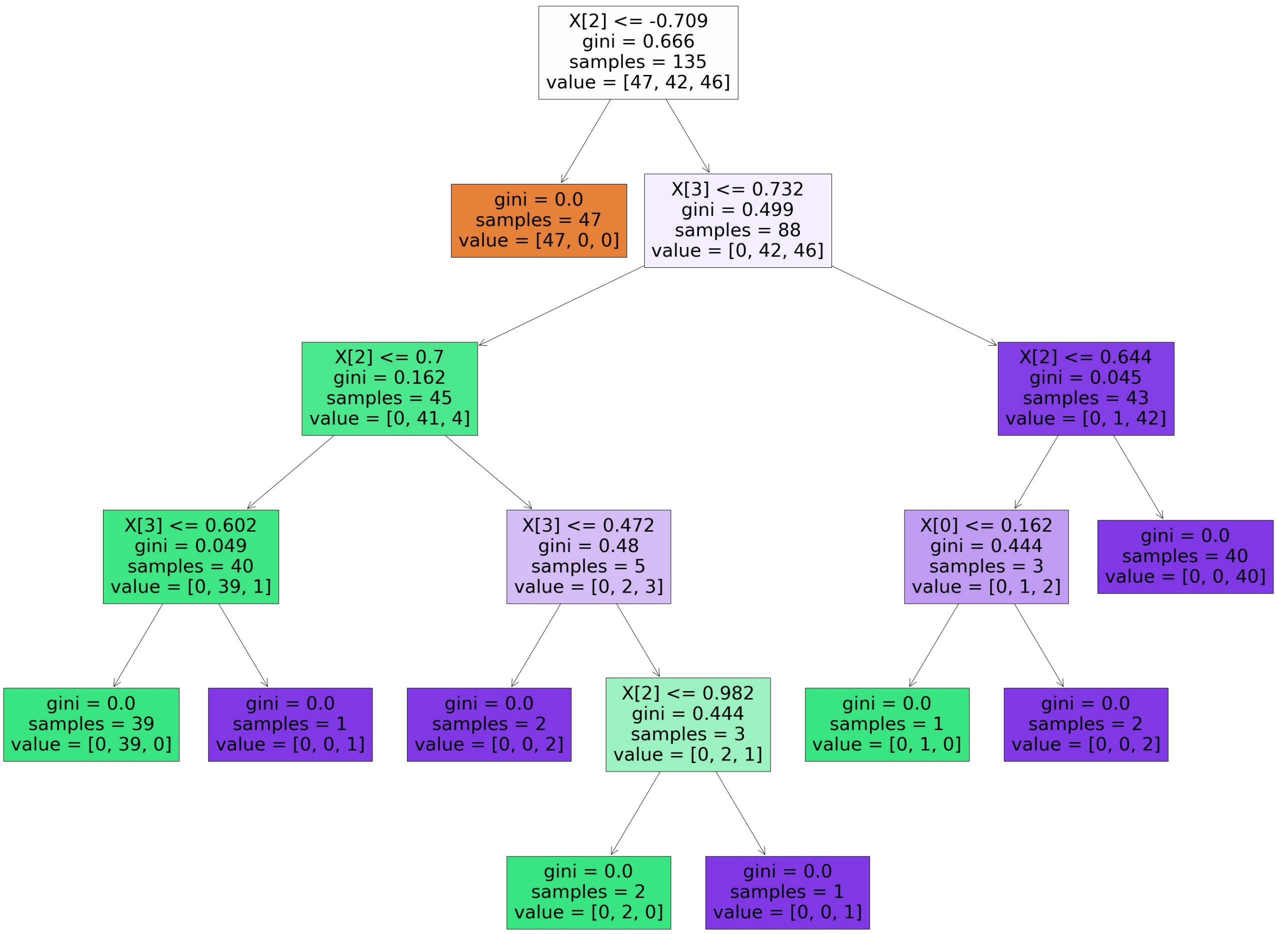
Plotting Decision Tree

In [117]:

```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
plt.figure(figsize=(50,40))
plot_tree(model,filled=True)
```

Out[117]:

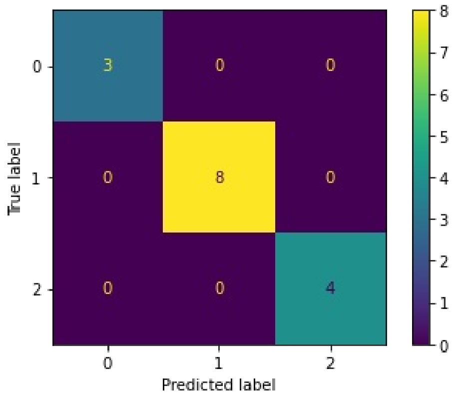
```
[Text(1395.0, 1993.2, 'X[2] <= -0.709\ngini = 0.666\nsamples = 135\nvalue = [47, 42, 46]'),
 Text(1180.3846153846155, 1630.8000000000002, 'gini = 0.0\nsamples = 47\nvalue = [47, 0, 0]'),
 Text(1609.6153846153845, 1630.8000000000002, 'X[3] <= 0.732\ngini = 0.499\nsamples = 88\nvalue = [0
, 42, 46]'),
 Text(858.4615384615385, 1268.4, 'X[2] <= 0.7\ngini = 0.162\nsamples = 45\nvalue = [0, 41, 4]'),
 Text(429.2307692307692, 906.0, 'X[3] <= 0.602\ngini = 0.049\nsamples = 40\nvalue = [0, 39, 1]'),
 Text(214.6153846153846, 543.5999999999999, 'gini = 0.0\nsamples = 39\nvalue = [0, 39, 0]'),
 Text(643.8461538461538, 543.5999999999999, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(1287.6923076923076, 906.0, 'X[3] <= 0.472\ngini = 0.48\nsamples = 5\nvalue = [0, 2, 3]'),
 Text(1073.076923076923, 543.5999999999999, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
 Text(1502.3076923076924, 543.5999999999999, 'X[2] <= 0.982\ngini = 0.444\nsamples = 3\nvalue = [0,
2, 1]'),
 Text(1287.6923076923076, 181.19999999999982, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
 Text(1716.923076923077, 181.19999999999982, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(2360.769230769231, 1268.4, 'X[2] <= 0.644\ngini = 0.045\nsamples = 43\nvalue = [0, 1, 42]'),
 Text(2146.153846153846, 906.0, 'X[0] <= 0.162\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),
 Text(1931.5384615384614, 543.5999999999999, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
 Text(2360.769230769231, 543.5999999999999, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
 Text(2575.3846153846152, 906.0, 'gini = 0.0\nsamples = 40\nvalue = [0, 0, 40]')]
```



Plotting Confusion Matrix

In [118]:

```
cfm = (metrics.plot_confusion_matrix(model,x_test,y_test))
```



Thank you for checking ...!!!

In [ ]: