In [1]:
```
import os
import numpy as np
import pandas as pd

from sklearn import model_selection
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import cohen_kappa_score
import seaborn as sns
```

In [2]:
```
!pip install matplotlib-scalebar
```

```
Requirement already satisfied: matplotlib-scalebar in c:\users\lenovo\anaconda3\lib
\site-packages (0.9.0)
Requirement already satisfied: matplotlib in c:\users\lenovo\anaconda3\lib\site-pack
ages (from matplotlib-scalebar) (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\lenovo\anaconda3\lib\sit
e-packages (from matplotlib->matplotlib-scalebar) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\lenovo\anaconda3\lib\site-pa
ckages (from matplotlib->matplotlib-scalebar) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\lenovo\anaconda3\lib\si
te-packages (from matplotlib->matplotlib-scalebar) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\lenovo\anaconda3\lib\si
te-packages (from matplotlib->matplotlib-scalebar) (1.4.4)
Requirement already satisfied: numpy>=1.23 in c:\users\lenovo\anaconda3\lib\site-pac
kages (from matplotlib->matplotlib-scalebar) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\lenovo\anaconda3\lib\site
-packages (from matplotlib->matplotlib-scalebar) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\lenovo\anaconda3\lib\site-packa
ges (from matplotlib->matplotlib-scalebar) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\lenovo\anaconda3\lib\sit
e-packages (from matplotlib->matplotlib-scalebar) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\lenovo\anaconda3\lib
\site-packages (from matplotlib->matplotlib-scalebar) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\lenovo\anaconda3\lib\site-packag
es (from python-dateutil>=2.7->matplotlib->matplotlib-scalebar) (1.16.0)
```

In [3]:
```
import rasterio as rio
import rasterio.plot
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from matplotlib_scalebar.scalebar import ScaleBar
import earthpy.plot as ep
from matplotlib.patches import Patch
```

In [4]:
```
os.getcwd()
```

Out[4]:
```
'C:\\Users\\lenovo'
```

In [5]:
```
os.chdir('E:/assignment/decision_tree')
```

In [8]:
```
data=pd.read_csv('sample_kolkata - sample_kolkata.csv')
data
```

Out[8]:

	B2	B3	B4	B5	B6	B7	MNDWI	NDBI	NDVI	LULC	SYM
0	9335	8513	8199	10561	10483	9265	-0.103706	-0.003707	0.125906	Built-up	
1	9473	8620	8467	10446	10632	9463	-0.104509	0.008824	0.104637	Built-up	
2	9354	8313	8154	9530	10483	9590	-0.115450	0.047619	0.077810	Built-up	
3	9458	8556	8830	9389	9895	9221	-0.072571	0.026239	0.030682	Built-up	
4	9697	8681	8535	10033	11246	10498	-0.128720	0.057005	0.080676	Built-up	
...	...	...	...	...	...	...	...	...	...	...	
2495	8652	7693	6811	6163	5321	5188	0.182265	-0.073319	-0.049946	water	
2496	8568	7635	6746	6121	5299	5173	0.180609	-0.071979	-0.048574	water	
2497	8614	7643	6794	6132	5344	5213	0.177023	-0.068665	-0.051215	water	
2498	8750	7892	7019	6298	5446	5285	0.183386	-0.072548	-0.054141	water	
2499	8751	7879	7028	6220	5454	5268	0.181880	-0.065616	-0.060990	water	

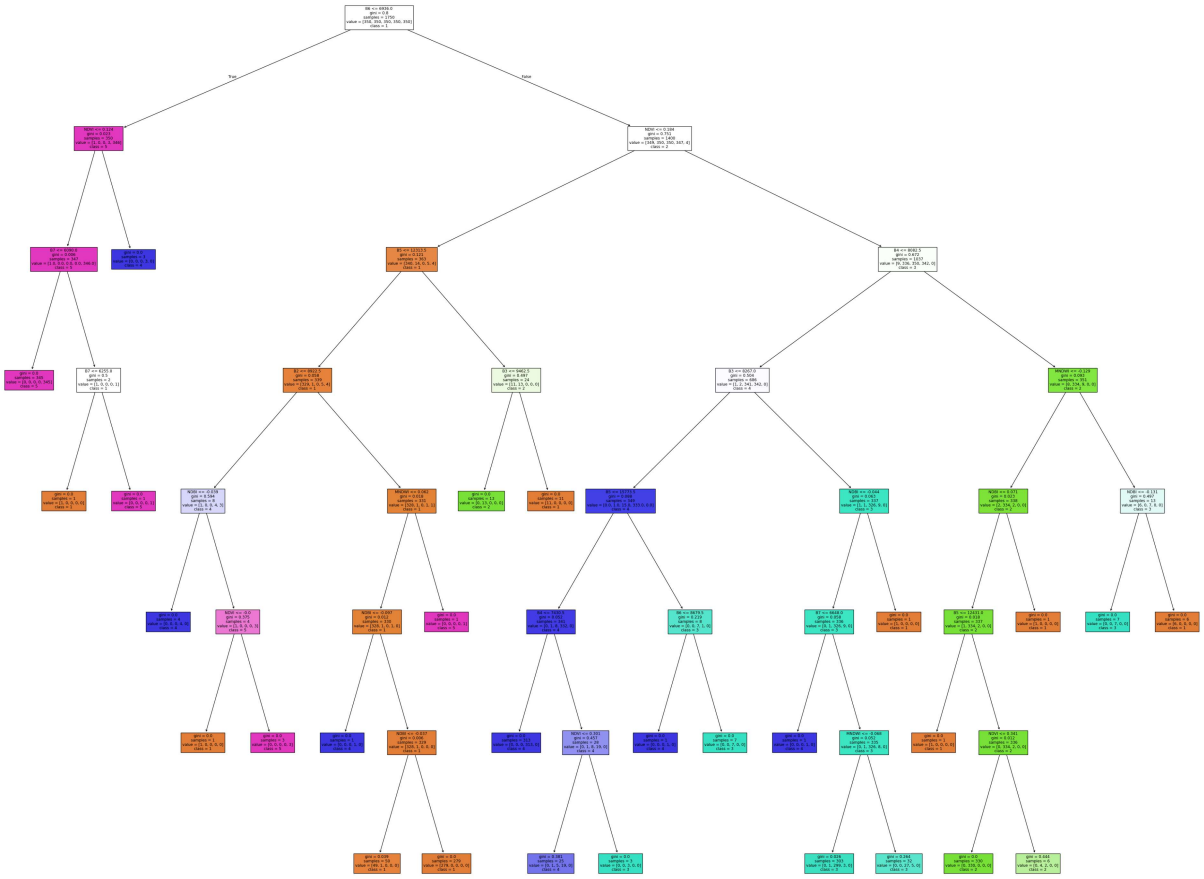
2500 rows × 11 columns

In [9]:
```
x=data.drop(['LULC','SYMBOL'],axis=1)
x=x[:].values
y=data.iloc[:,-1]
y=y[:].values
```

In [10]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.30,stratify=y, ran
```

In [11]:
```
DT=DecisionTreeClassifier(criterion='gini',random_state=10,max_depth=7)
DT.fit(x_train,y_train)
y_train_pred=DT.predict(x_train)
y_test_pred=DT.predict(x_test)
```

In [12]:
```
plt.figure (figsize=(50,40))
classes=['1','2','3','4','5']
features=['B2','B3','B4','B5','B6','B7','MNDWI','NDBI','NDVI']
plot_tree(DT,feature_names=features,class_names=classes,filled=True)
plt.show()
```



In [13]:
```
print('Train score: ', accuracy_score(y_train_pred,y_train))
print('Test score: ', accuracy_score(y_test_pred,y_test))
```

```
Train score:  0.9897142857142858
Test score:  0.9773333333333334
```

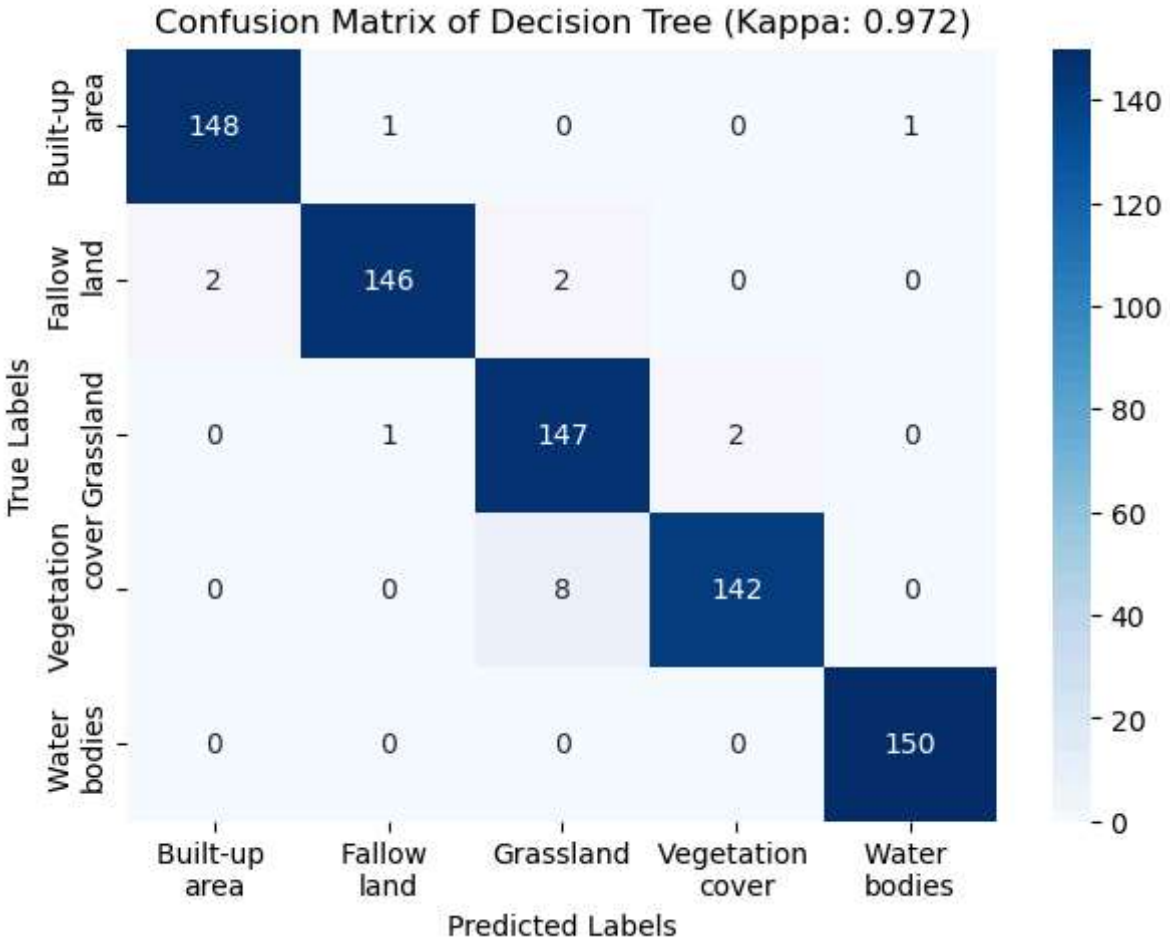
In [14]:
```
CM_DT=confusion_matrix(y_test,y_test_pred)
CM_DT
```

Out[14]:
```
array([[148,   1,   0,   0,   1],
       [  2, 146,   2,   0,   0],
       [  0,   1, 147,   2,   0],
       [  0,   0,   8, 142,   0],
       [  0,   0,   0,   0, 150]], dtype=int64)
```

In [15]:
```
# Define class names corresponding to numerical labels
class_labels = {
    0: 'Built-up \narea',
    1: 'Fallow \nland',
    2: 'Grassland',
    3: 'Vegetation \ncover',
    4: 'Water \nbodies'}

# Compute confusion matrix
CM_CTB_best = confusion_matrix(y_test, y_test_pred)

# Compute Cohen's Kappa coefficient
```

```
kappaCTB = cohen_kappa_score(y_test, y_test_pred)
print(f"Cohen's Kappa Score: {kappaCTB:.3f}")

# Replace numeric labels with text labels
tick_labels = [class_labels[i] for i in sorted(class_labels.keys())]

# Plot confusion matrix using seaborn
plt.figure(figsize=(7, 5))
sns.heatmap(CM_CTB_best, annot=True, fmt='d', cmap='Blues', xticklabels=tick_labels
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title(f"Confusion Matrix of Decision Tree (Kappa: {kappaCTB:.3f})")
#plt.savefig('CTBConfMat.jpg', dpi=1000, format='jpg')
plt.show()
```

```
Cohen's Kappa Score: 0.972
```

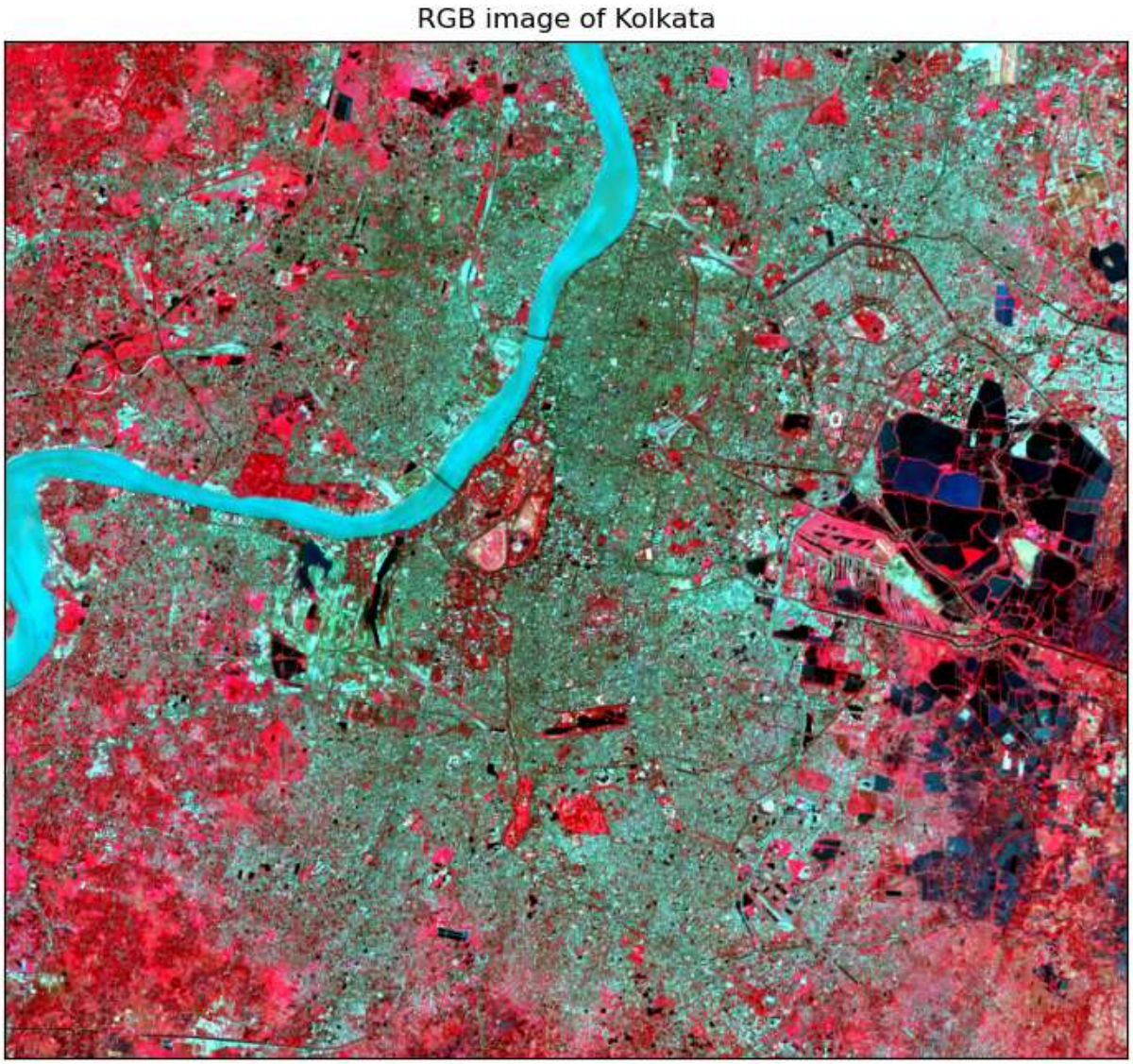


In [16]:
```
print(classification_report(y_test,y_test_pred))
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	150
1	0.99	0.97	0.98	150
2	0.94	0.98	0.96	150
3	0.99	0.95	0.97	150
4	0.99	1.00	1.00	150
accuracy			0.98	750
macro avg	0.98	0.98	0.98	750
weighted avg	0.98	0.98	0.98	750

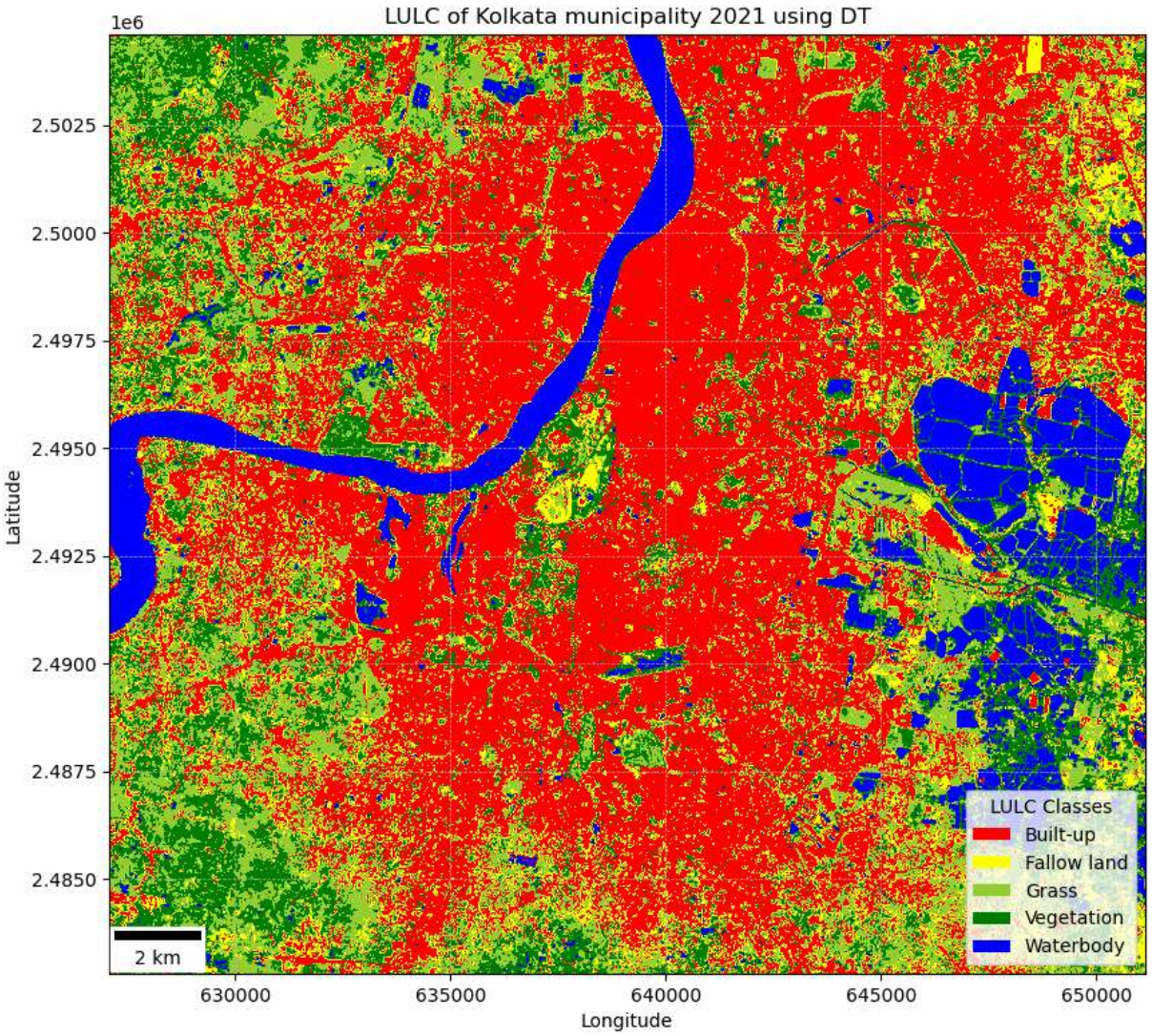
In [17]:
```
img=rio.open('compkol.tif')
img=img.read()
```

In [18]:
```
ep.plot_rgb(img,rgb=[3,2,1],stretch=True,figsize=(10,8),title='RGB image of Kolkata
plt.show()
```



In [19]:
```
with rio.open('compkol.tif') as src:
    img = src.read()
```

```
    transform = src.transform
    crs = src.crs
    extent = rasterio.plot.plotting_extent(src)


Image = img.transpose([1, 2, 0])
new_Image = np.nan_to_num(Image)
Bands = new_Image.reshape(Image.shape[0]*Image.shape[1], Image.shape[2])


Pred_DT_Img = DT.predict(Bands)
Pred_DT_Img = Pred_DT_Img.reshape(Image.shape[0], Image.shape[1])


cmap = ListedColormap(['red', 'yellow', 'yellowgreen', 'green', 'blue'])
class_labels = {
    0: "Built-up",
    1: "Fallow land",
    2: "Grass",
    3: "Vegetation",
    4: "Waterbody"
}


fig, ax = plt.subplots(figsize=(10, 12))

im = ax.imshow(Pred_DT_Img, cmap=cmap, extent=extent, origin="upper")


scalebar = ScaleBar(dx=1, units="m", location='lower left')
ax.add_artist(scalebar)


ax.set_xlabel("Longitude")
ax.set_ylabel("Latitude")
ax.set_title("LULC of Kolkata municipality 2021 using DT")
ax.grid(True, linestyle="--", linewidth=0.5)

legend_elements = [Patch(facecolor=cmap(i), label=label)
                   for i, label in class_labels.items()]
ax.legend(handles=legend_elements,
          loc='lower right',
          title="LULC Classes")

plt.show()
```


LULC of Kolkata municipality 2021 using DT
1e6
2.5025
2.5000
2.4975
2.4950
2.4925
2.4900
2.4875
2.4850
Latitude
630000
635000
640000
645000
650000
Longitude
2 km
LULC Classes
Built-up
Fallow land
Grass
Vegetation
Waterbody

In [ ]:

In [ ]: