In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import re
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.tree import plot_tree
```

In [3]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\C6_bmi - C6_bmi.csv")
df
```

Out[3]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

In [6]:
```
df1=df.fillna(value=0)
df1
```

Out[6]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

In [7]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Gender  500 non-null    object
 1   Height  500 non-null    int64
 2   Weight  500 non-null    int64
 3   Index   500 non-null    int64
dtypes: int64(3), object(1)
memory usage: 15.8+ KB
```

In [8]:
```
df1.columns
```

Out[8]:
```
Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')
```

In [19]:
```
df2=df1[[ 'Height', 'Weight', 'Index']]
df2
```

Out[19]:

	Height	Weight	Index
0	174	96	4
1	189	87	2
2	185	110	4
3	195	104	3
4	149	61	3
...	...	...	...
495	150	153	5
496	184	121	4
497	141	136	5
498	150	95	5
499	173	131	5

500 rows × 3 columns

In [20]:
```
df2['Index'].value_counts()
```

Out[20]:
```
5    198
4    130
2     69
3     68
1     22
0     13
Name: Index, dtype: int64
```

In [21]:
```
x=df2.drop('Index',axis=1)
y=df2['Index']
```

In [22]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.70)
```

In [23]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[23]:
```
RandomForestClassifier()
```

In [24]:
```
parameters = {'max_depth':[1,2,3,4,5],
              'min_samples_leaf':[5,10,15,20,25],
               'n_estimators':[10,20,30,40,50]}
```

In [25]:
```
grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='ac
grid_search.fit(x_train,y_train)
```

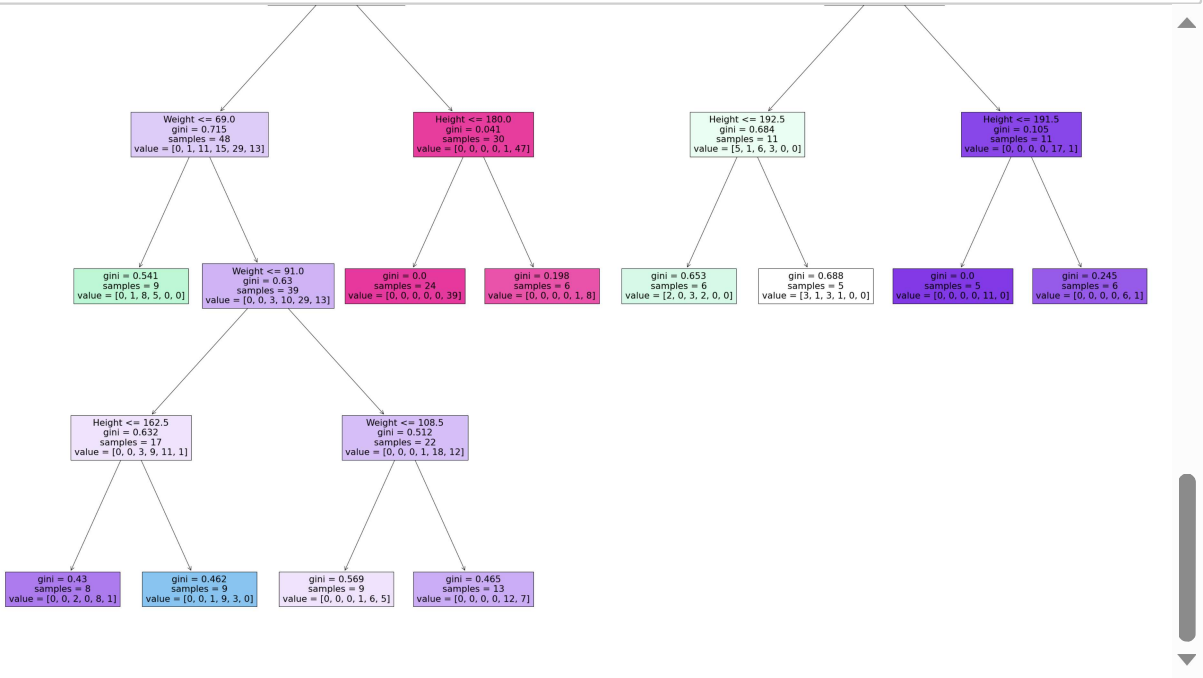
Out[25]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 10, 15, 20, 25],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [26]:
```
grid_search.best_score_
```

Out[26]:
```
0.7
```

In [27]:
```
rfc_best =grid_search.best_estimator_
```

In [30]:
```
plt.figure(figsize=(50,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,filled=True)
```



In [ ]:
```
```