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 [2]:
```
df=pd.read_csv("c7_used_cars - c7_used_cars.csv")
df
```

Out[2]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	M
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2.0	
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2.0	
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2.0	
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2.0	
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1.5	
...	...	...	...	...	...	...	...	...	...	...	
99182	10663	A3	2020	16999	Manual	4018	Petrol	145	49.6	1.0	
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1.0	
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1.0	
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1.4	
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1.4	

99187 rows × 11 columns

In [3]:
```
df1=df.fillna(value=0)
df1
```

Out[3]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	M
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2.0	
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2.0	
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2.0	
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2.0	
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1.5	
...	...	...	...	...	...	...	...	...	...	...	
99182	10663	A3	2020	16999	Manual	4018	Petrol	145	49.6	1.0	
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1.0	
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1.0	
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1.4	
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1.4	

99187 rows × 11 columns

In [4]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99187 entries, 0 to 99186
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    99187 non-null  int64
 1   model         99187 non-null  object
 2   year          99187 non-null  int64
 3   price         99187 non-null  int64
 4   transmission  99187 non-null  object
 5   mileage       99187 non-null  int64
 6   fuelType      99187 non-null  object
 7   tax           99187 non-null  int64
 8   mpg           99187 non-null  float64
 9   engineSize    99187 non-null  float64
 10  Make          99187 non-null  object
dtypes: float64(2), int64(5), object(4)
memory usage: 8.3+ MB
```

In [5]:
```
df1.columns
```

Out[5]:
```
Index(['Unnamed: 0', 'model', 'year', 'price', 'transmission', 'mileage',
       'fuelType', 'tax', 'mpg', 'engineSize', 'Make'],
      dtype='object')
```

In [6]:
```
df2=df1[['Unnamed: 0','year', 'price', 'mileage','tax', 'mpg', 'engineSize', 'M
df2
```

Out[6]:

	Unnamed: 0	year	price	mileage	tax	mpg	engineSize	Make
0	0	2019	25000	13904	145	49.6	2.0	VW
1	1	2019	26883	4562	145	49.6	2.0	VW
2	2	2019	20000	7414	145	50.4	2.0	VW
3	3	2019	33492	4825	145	32.5	2.0	VW
4	4	2019	22900	6500	150	39.8	1.5	VW
...	...	...	...	...	...	...	...	...
99182	10663	2020	16999	4018	145	49.6	1.0	Audi
99183	10664	2020	16999	1978	150	49.6	1.0	Audi
99184	10665	2020	17199	609	150	49.6	1.0	Audi
99185	10666	2017	19499	8646	150	47.9	1.4	Audi
99186	10667	2016	15999	11855	150	47.9	1.4	Audi

99187 rows × 8 columns

In [8]:
```
df2['Make'].value_counts()
```

Out[8]:
```
ford          17965
VW            15157
vauxhall      13632
merc          13119
BMW           10781
Audi          10668
toyota         6738
skoda          6267
hyundi         4860
Name: Make, dtype: int64
```

In [10]:
```
x=df2.drop('Make',axis=1)
y=df2['Make']
```

In [11]:
```
g1={"Make":{'VW':1,"Audi":2}}
df2=df2.replace(g1)
print(df2)
```

```
       Unnamed: 0  year  price  mileage  tax   mpg  engineSize Make
0               0  2019  25000    13904  145  49.6         2.0    1
1               1  2019  26883     4562  145  49.6         2.0    1
2               2  2019  20000     7414  145  50.4         2.0    1
3               3  2019  33492     4825  145  32.5         2.0    1
4               4  2019  22900     6500  150  39.8         1.5    1
...           ...   ...    ...      ...  ...   ...         ...  ...
99182       10663  2020  16999     4018  145  49.6         1.0    2
99183       10664  2020  16999     1978  150  49.6         1.0    2
99184       10665  2020  17199      609  150  49.6         1.0    2
99185       10666  2017  19499     8646  150  47.9         1.4    2
99186       10667  2016  15999    11855  150  47.9         1.4    2

[99187 rows x 8 columns]
```

In [12]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.70)
```

In [13]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[13]:
```
RandomForestClassifier()
```

In [14]:
```
parameters = {'max_depth':[1,2,3,4,5],
              'min_samples_leaf':[5,10,15,20,25],
               'n_estimators':[10,20,30,40,50]}
```

In [15]:
```
grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='ac
grid_search.fit(x_train,y_train)
```

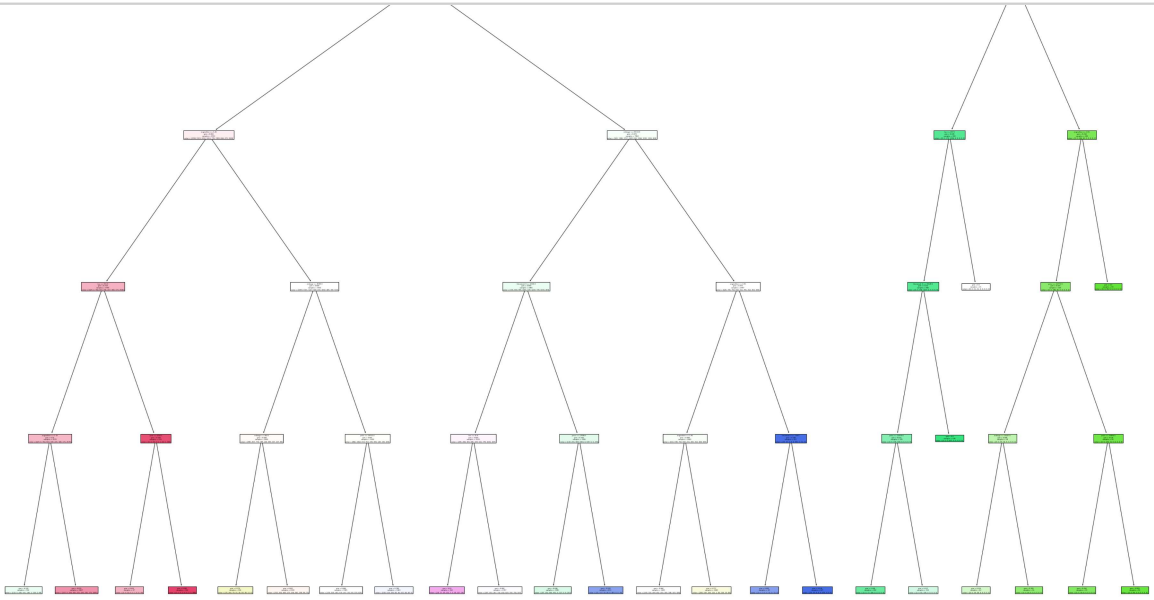
Out[15]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 10, 15, 20, 25],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [16]:
```
grid_search.best_score_
```

Out[16]:
```
0.5086033068960882
```

In [17]:
```
rfc_best =grid_search.best_estimator_
```

In [20]:
```
plt.figure(figsize=(50,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,filled=True)
```



In [ ]: