In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df=pd.read_csv(r"c7_used_cars.csv")
df
```

Out[3]:

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

99187 rows × 11 columns

In [4]:
```
df.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]:
```
df=df.dropna()
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Unnamed: 0      0
model           0
year            0
price           0
transmission    0
mileage         0
fuelType        0
tax             0
mpg             0
engineSize      0
Make            0
dtype: int64
```

In [7]:
```
df.describe()
```

Out[7]:

| | Unnamed: 0 | year | price | mileage | tax | mpg | engineSize |
|---|---|---|---|---|---|---|---|
| **count** | 99187.000000 | 99187.000000 | 99187.000000 | 99187.000000 | 99187.000000 | 99187.000000 | 99187.000000 |
| **mean** | 6294.413532 | 2017.087723 | 16805.347656 | 23058.914213 | 120.299838 | 55.166825 | 1.663280 |
| **std** | 4265.588536 | 2.123934 | 9866.773417 | 21148.523721 | 63.150926 | 16.138522 | 0.557646 |
| **min** | 0.000000 | 1970.000000 | 450.000000 | 1.000000 | 0.000000 | 0.300000 | 0.000000 |
| **25%** | 2755.000000 | 2016.000000 | 9999.000000 | 7425.000000 | 125.000000 | 47.100000 | 1.200000 |
| **50%** | 5591.000000 | 2017.000000 | 14495.000000 | 17460.000000 | 145.000000 | 54.300000 | 1.600000 |
| **75%** | 9420.000000 | 2019.000000 | 20870.000000 | 32339.000000 | 145.000000 | 62.800000 | 2.000000 |
| **max** | 17964.000000 | 2060.000000 | 159999.000000 | 323000.000000 | 580.000000 | 470.800000 | 6.600000 |

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

Out[8]:
```
Index(['Unnamed: 0', 'model', 'year', 'price', 'transmission', 'mileage',
       'fuelType', 'tax', 'mpg', 'engineSize', 'Make'],
      dtype='object')
```

In [12]:
```
df['transmission'].value_counts()
```

Out[12]:
```
Manual       56445
Semi-Auto    22677
Automatic    20056
Other            9
Name: transmission, dtype: int64
```

In [13]:
```
g1={"transmission":{'Other':0,'Manual':1,'Semi-Auto':2,'Automatic':3}}
df=df.replace(g1)
print(df)
```

```
       Unnamed: 0   model  year  price  transmission  mileage fuelType  tax  \
0               0   T-Roc  2019  25000             3    13904   Diesel  145
1               1   T-Roc  2019  26883             3     4562   Diesel  145
2               2   T-Roc  2019  20000             1     7414   Diesel  145
3               3   T-Roc  2019  33492             3     4825   Petrol  145
4               4   T-Roc  2019  22900             2     6500   Petrol  150
...           ...     ...   ...    ...           ...      ...      ...  ...
99182       10663      A3  2020  16999             1     4018   Petrol  145
99183       10664      A3  2020  16999             1     1978   Petrol  150
99184       10665      A3  2020  17199             1      609   Petrol  150
99185       10666      Q3  2017  19499             3     8646   Petrol  150
99186       10667      Q3  2016  15999             1    11855   Petrol  150

        mpg  engineSize  Make
0      49.6         2.0    VW
1      49.6         2.0    VW
2      50.4         2.0    VW
3      32.5         2.0    VW
4      39.8         1.5    VW
...     ...         ...   ...
99182  49.6         1.0  Audi
99183  49.6         1.0  Audi
99184  49.6         1.0  Audi
99185  47.9         1.4  Audi
99186  47.9         1.4  Audi

[99187 rows x 11 columns]
```

In [16]:
```
x=df[['Unnamed: 0','year','price','mileage','tax','mpg','engineSize']]
y=df["transmission"]
```

In [17]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [18]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[18]:
```
RandomForestClassifier()
```

In [19]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]}
```

In [20]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[20]:
```
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 [21]:
```
grid_search.best_score_
```

Out[21]:
```
0.7316289788275961
```

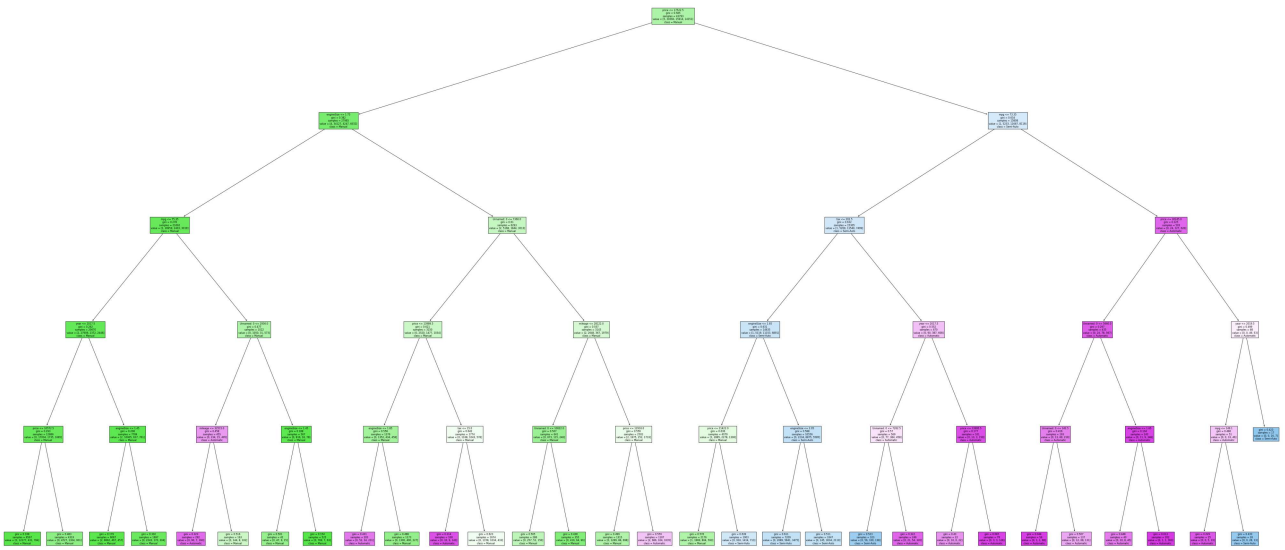
In [22]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]}
```

In [23]:
```
rfc_best=grid_search.best_estimator_
```

In [24]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Other','Manual','Semi-Auto','Automatic
```

Out[24]:
[Text(2332.622950819672, 1993.2, 'price <= 17522.5\ngini = 0.585\nsamples = 43793\nvalue = [5, 39360, 15914, 14151]\nclass = Manual'),
 Text(1170.8852459016393, 1630.8000000000002, 'engineSize <= 1.75\ngini = 0.382\nsamples = 27985\nvalue = [4, 34127, 4247, 6032]\nclass = Manual'),
 Text(585.4426229508197, 1268.4, 'mpg <= 75.35\ngini = 0.278\nsamples = 21692\nvalue = [2, 28959, 2403, 3019]\nclass = Manual'),
 Text(292.72131147540983, 906.0, 'year <= 2017.5\ngini = 0.262\nsamples = 20670\nvalue = [2, 27909, 2372, 2446]\nclass = Manual'),
 Text(146.36065573770492, 543.5999999999999, 'price <= 10572.5\ngini = 0.293\nsamples = 12886\nvalue = [0, 17004, 1735, 1685]\nclass = Manual'),
 Text(73.18032786885246, 181.19999999999982, 'gini = 0.168\nsamples = 8567\nvalue = [0, 12277, 431, 784]\nclass = Manual'),
 Text(219.54098360655738, 181.19999999999982, 'gini = 0.483\nsamples = 4319\nvalue = [0, 4727, 1304, 901]\nclass = Manual'),
 Text(439.08196721311475, 543.5999999999999, 'engineSize <= 1.45\ngini = 0.208\nsamples = 7784\nvalue = [2, 10905, 637, 761]\nclass = Manual'),
 Text(365.9016393442623, 181.19999999999982, 'gini = 0.179\nsamples = 6087\nvalue = [2, 8662, 467, 457]\nclass = Manual'),
 Text(512.2622950819672, 181.19999999999982, 'gini = 0.302\nsamples = 1697\nvalue = [0, 2243, 170, 304]\nclass = Manual'),
 Text(878.1639344262295, 906.0, 'Unnamed: 0 <= 2936.0\ngini = 0.477\nsamples = 1022\nvalue = [0, 1050, 31, 573]\nclass = Manual'),
 Text(731.8032786885246, 543.5999999999999, 'mileage <= 32515.0\ngini = 0.458\nsamples = 455\nvalue = [0, 234, 15, 495]\nclass = Automatic'),
 Text(658.6229508196722, 181.19999999999982, 'gini = 0.323\nsamples = 292\nvalue = [0, 90, 7, 393]\nclass = Automatic'),
 Text(804.983606557377, 181.19999999999982, 'gini = 0.516\nsamples = 163\nvalue = [0, 144, 8, 102]\nclass = Manual'),
 Text(1024.5245901639344, 543.5999999999999, 'engineSize <= 1.45\ngini = 0.188\nsamples = 567\nvalue = [0, 816, 16, 78]\nclass = Manual'),
 Text(951.344262295082, 181.19999999999982, 'gini = 0.501\nsamples = 42\nvalue = [0, 47, 9, 15]\nclass = Manual'),
 Text(1097.704918032787, 181.19999999999982, 'gini = 0.154\nsamples = 525\nvalue = [0, 769, 7, 63]\nclass = Manual'),
 Text(1756.327868852459, 1268.4, 'Unnamed: 0 <= 7360.0\ngini = 0.61\nsamples = 6293\nvalue = [2, 5168, 1844, 3013]\nclass = Manual'),
 Text(1463.6065573770493, 906.0, 'price <= 13989.5\ngini = 0.621\nsamples = 3150\nvalue = [0, 2520, 1477, 1034]\nclass = Manual'),
 Text(1317.2459016393443, 543.5999999999999, 'engineSize <= 1.85\ngini = 0.558\nsamples = 1376\nvalue = [0, 1352, 434, 458]\nclass = Manual'),
 Text(1244.0655737704917, 181.19999999999982, 'gini = 0.431\nsamples = 203\nvalue = [0, 52, 34, 231]\nclass = Automatic'),
 Text(1390.4262295081967, 181.19999999999982, 'gini = 0.488\nsamples = 1173\nvalue = [0, 1300, 400, 227]\nclass = Manual'),
 Text(1609.967213114754, 543.5999999999999, 'tax <= 15.0\ngini = 0.642\nsamples = 1774\nvalue = [0, 1168, 1043, 576]\nclass = Manual'),
 Text(1536.7868852459017, 181.19999999999982, 'gini = 0.214\nsamples = 100\nvalue = [0, 10, 9, 143]\nclass = Automatic'),
 Text(1683.1475409836066, 181.19999999999982, 'gini = 0.623\nsamples = 1674\nvalue = [0, 1158, 1034, 433]\nclass = Manual'),
 Text(2049.0491803278687, 906.0, 'mileage <= 28122.0\ngini = 0.557\nsamples = 3143\nvalue = [2, 2648, 367, 1979]\nclass = Manual'),
 Text(1902.688524590164, 543.5999999999999, 'Unnamed: 0 <= 10822.0\ngini = 0.507\nsamples = 641\nvalue = [0, 673, 115, 246]\nclass = Manual'),
 Text(1829.5081967213114, 181.19999999999982, 'gini = 0.565\nsamples = 286\nvalue = [0, 257, 51, 150]\nclass = Manual'),
 Text(1975.8688524590164, 181.19999999999982, 'gini = 0.438\nsamples = 355\nvalue = [0, 416, 64, 96]\nclass = Manual'),
 Text(2195.409836065574, 543.5999999999999, 'price <= 12916.0\ngini = 0.556\nsamples = 2502\nvalue = [2, 1975, 252, 1733]\nclass = Manual'),
 Text(2122.2295081967213, 181.19999999999982, 'gini = 0.499\nsamples = 1315\nvalue = [0, 1289, 88, 698]\nclass = Manual'),
 Text(2268.590163934426, 181.19999999999982, 'gini = 0.559\nsamples = 1187\nvalue = [2, 686, 164, 1035]\nclass = Automatic'),
 Text(3494.3606557377047, 1630.8000000000002, 'mpg <= 73.35\ngini = 0.634\nsamples = 15808\nvalue = [1, 5233, 11667, 8119]\nclass = Semi-Auto'),
 Text(2927.2131147540986, 1268.4, 'tax <= 202.5\ngini = 0.632\nsamples = 15305\nvalue = [1, 5209, 11540, 7499]\nclass = Semi-Auto'),
 Text(2634.4918032786886, 906.0, 'engineSize <= 1.65\ngini = 0.631\nsamples = 14635\nvalue = [1, 5119, 11153, 6891]\nclass = Semi-Auto'),
 Text(2488.1311475409834, 543.5999999999999, 'price <= 21422.0\ngini = 0.636\nsamples = 4079\nvalue = [1, 2885, 2278, 1306]\nclass = Manual'),
 Text(2414.9508196721313, 181.19999999999982, 'gini = 0.576\nsamples = 2176\nvalue = [1, 1969, 864, 593]\nclass = Manual'),
 Text(2561.311475409836, 181.19999999999982, 'gini = 0.639\nsamples = 1903\nvalue = [0, 916, 1414, 713]\nclas

```
s = Semi-Auto'),
 Text(2780.8524590163934, 543.5999999999999, 'engineSize <= 2.05\ngini = 0.588\nsamples = 10556\nvalue = [0,
2234, 8875, 5585]\nclass = Semi-Auto'),
 Text(2707.6721311475408, 181.19999999999982, 'gini = 0.611\nsamples = 7209\nvalue = [0, 2089, 5841, 3475]\nc
lass = Semi-Auto'),
 Text(2854.032786885246, 181.19999999999982, 'gini = 0.511\nsamples = 3347\nvalue = [0, 145, 3034, 2110]\ncla
ss = Semi-Auto'),
 Text(3219.934426229508, 906.0, 'year <= 2017.5\ngini = 0.552\nsamples = 670\nvalue = [0, 90, 387, 608]\nclas
s = Automatic'),
 Text(3073.5737704918033, 543.5999999999999, 'Unnamed: 0 <= 7262.5\ngini = 0.57\nsamples = 569\nvalue = [0, 7
7, 384, 458]\nclass = Automatic'),
 Text(3000.3934426229507, 181.19999999999982, 'gini = 0.523\nsamples = 323\nvalue = [0, 56, 330, 138]\nclass
= Semi-Auto'),
 Text(3146.754098360656, 181.19999999999982, 'gini = 0.322\nsamples = 246\nvalue = [0, 21, 54, 320]\nclass =
Automatic'),
 Text(3366.2950819672133, 543.5999999999999, 'price <= 23995.5\ngini = 0.177\nsamples = 101\nvalue = [0, 13,
3, 150]\nclass = Automatic'),
 Text(3293.1147540983607, 181.19999999999982, 'gini = 0.43\nsamples = 22\nvalue = [0, 10, 0, 22]\nclass = Aut
omatic'),
 Text(3439.4754098360654, 181.19999999999982, 'gini = 0.087\nsamples = 79\nvalue = [0, 3, 3, 128]\nclass = Au
tomatic'),
 Text(4061.5081967213114, 1268.4, 'price <= 30245.0\ngini = 0.325\nsamples = 503\nvalue = [0, 24, 127, 620]\n
class = Automatic'),
 Text(3805.377049180328, 906.0, 'Unnamed: 0 <= 5660.0\ngini = 0.267\nsamples = 435\nvalue = [0, 24, 78, 567]
\nclass = Automatic'),
 Text(3659.0163934426228, 543.5999999999999, 'Unnamed: 0 <= 342.5\ngini = 0.416\nsamples = 193\nvalue = [0, 1
3, 69, 219]\nclass = Automatic'),
 Text(3585.8360655737706, 181.19999999999982, 'gini = 0.044\nsamples = 56\nvalue = [0, 1, 1, 88]\nclass = Aut
omatic'),
 Text(3732.1967213114754, 181.19999999999982, 'gini = 0.507\nsamples = 137\nvalue = [0, 12, 68, 131]\nclass =
Automatic'),
 Text(3951.7377049180327, 543.5999999999999, 'engineSize <= 1.65\ngini = 0.104\nsamples = 242\nvalue = [0, 1
1, 9, 348]\nclass = Automatic'),
 Text(3878.55737704918, 181.19999999999982, 'gini = 0.448\nsamples = 40\nvalue = [0, 10, 8, 45]\nclass = Auto
matic'),
 Text(4024.9180327868853, 181.19999999999982, 'gini = 0.013\nsamples = 202\nvalue = [0, 1, 1, 303]\nclass = A
utomatic'),
 Text(4317.639344262295, 906.0, 'year <= 2019.5\ngini = 0.499\nsamples = 68\nvalue = [0, 0, 49, 53]\nclass =
Automatic'),
 Text(4244.459016393443, 543.5999999999999, 'mpg <= 149.1\ngini = 0.486\nsamples = 51\nvalue = [0, 0, 33, 46]
\nclass = Automatic'),
 Text(4171.2786885245905, 181.19999999999982, 'gini = 0.229\nsamples = 25\nvalue = [0, 0, 5, 33]\nclass = Aut
omatic'),
 Text(4317.639344262295, 181.19999999999982, 'gini = 0.433\nsamples = 26\nvalue = [0, 0, 28, 13]\nclass = Sem
i-Auto'),
 Text(4390.819672131148, 543.5999999999999, 'gini = 0.423\nsamples = 17\nvalue = [0, 0, 16, 7]\nclass = Semi-
Auto')]
```



In [ ]: