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

```
In [3]: df=pd.read_csv(r"C:\Users\user\Downloads\c7_used_cars - c7_used_cars.csv")
        df
```

Out[3]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	Ma
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	A
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1.0	A
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1.0	A
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1.4	A
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1.4	A

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.columns
```

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

```
In [6]: df['Make'].value_counts()
```

Out[6]: ford   17965
        VW   15157
        vauxhall   13632
        merc   13119
        BMW   10781
        Audi   10668
        toyota   6738
        skoda   6267
        hyundi   4860
        Name: Make, dtype: int64

```
In [7]: df1=df[['Unnamed: 0', 'year', 'price', 'mileage','tax', 'mpg', 'engineSize', 'Mak
```

```
In [8]: x=df1.drop('Make',axis=1)
        y=df1['Make']
```

```
In [9]: g1={'Make':{'ford':1,"VW":2,"vauxhall":3,'merc':4,'BMW':5,'Audi':6,'toyota':7,'sk
        df1=df1.replace(g1)
        print(df)
```

```
        Unnamed: 0  model  year  price  transmission  mileage  fuelType  tax  \
0                0  T-Roc  2019  25000     Automatic    13904    Diesel  145
1                1  T-Roc  2019  26883     Automatic     4562    Diesel  145
2                2  T-Roc  2019  20000        Manual     7414    Diesel  145
3                3  T-Roc  2019  33492     Automatic     4825    Petrol  145
4                4  T-Roc  2019  22900     Semi-Auto     6500    Petrol  150
...            ...    ...   ...    ...           ...      ...       ...  ...
99182        10663     A3  2020  16999        Manual     4018    Petrol  145
99183        10664     A3  2020  16999        Manual     1978    Petrol  150
99184        10665     A3  2020  17199        Manual      609    Petrol  150
99185        10666     Q3  2017  19499     Automatic     8646    Petrol  150
99186        10667     Q3  2016  15999        Manual    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 [10]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=45)
```

Random Forest

```
In [11]: from sklearn.ensemble import RandomForestClassifier
         rfc = RandomForestClassifier()
         rfc.fit(x_train,y_train)
```

```
Out[11]: RandomForestClassifier()
```

```
In [12]: parameters = {'max_depth':[1,2,3,4,5],
             'min_samples_leaf':[5,10,15,20,25],
             'n_estimators':[10,20,30,40,50]}
```

```
In [13]:  from sklearn.model_selection import GridSearchCV

          grid_search =  GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='acc
          grid_search.fit(x_train,y_train)
```

Out[13]: 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 [14]:  grid_search.best_score_
```

Out[14]: 0.5148675636965161

```
In [15]:  rfc_best = grid_search.best_estimator_
```

```
In [17]:  from sklearn.tree import plot_tree

          plt.figure(figsize=(80,40))
          plot_tree(rfc_best.estimators_[5],feature_names=x.columns,filled=True)
```

Out[17]: [Text(2332.622950819672, 1993.2, 'engineSize <= 1.95\ngini = 0.873\nsamples =
          62793\nvalue = [10643, 10707, 15413, 18084, 4860, 12994, 6182, 6702\n1355
          7]'),
           Text(1170.8852459016393, 1630.8000000000002, 'mpg <= 56.0\ngini = 0.844\nsam
          ples = 38153\nvalue = [4024, 1583, 9220, 14523, 4242, 3371, 4579, 5816, 1291
          0]'),
           Text(585.4426229508197, 1268.4, 'year <= 2018.5\ngini = 0.814\nsamples = 193
          29\nvalue = [2530, 1017, 4724, 5169, 1859, 2024, 1827, 1135, 10267]'),
           Text(292.72131147540983, 906.0, 'tax <= 197.5\ngini = 0.766\nsamples = 11231
          \nvalue = [1281, 501, 1715, 3831, 1087, 671, 825, 674, 7170]'),
           Text(146.36065573770492, 543.5999999999999, 'price <= 13338.0\ngini = 0.782
          \nsamples = 10190\nvalue = [1244, 498, 1659, 3597, 1042, 618, 819, 592, 607
          6]'),
           Text(73.18032786885246, 181.19999999999982, 'gini = 0.684\nsamples = 7308\nv
          alue = [263, 216, 800, 2955, 743, 101, 398, 451, 5652]'),
           Text(219.54098360655738, 181.19999999999982, 'gini = 0.86\nsamples = 2882\nv
          alue = [981, 282, 859, 642, 299, 517, 421, 141, 424]'),
           Text(439.08196721311475, 543.5999999999999, 'Unnamed: 0 <= 13437.5\ngini =
          0.511\nsamples = 1041\nvalue = [37, 3, 56, 234, 45, 53, 6, 82, 1094]'),
```

```
In [ ]:
```