

Random Forest

```
In [4]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [9]: df1=pd.read_csv(r"C:\Users\user\Downloads\c7_used_cars.csv")
df1
```

Out[9]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
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 [10]: df1.columns
```

```
Out[10]: Index(['Unnamed: 0', 'model', 'year', 'price', 'transmission', 'mileage',
               'fuelType', 'tax', 'mpg', 'engineSize', 'Make'],
              dtype='object')
```

In [11]: 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 [12]: df=df1.head(10)
df

Out[12]:

	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
5	5	T-Roc	2020	31895	Manual	10	Petrol	145	42.2	1.5	VW
6	6	T-Roc	2020	27895	Manual	10	Petrol	145	42.2	1.5	VW
7	7	T-Roc	2020	39495	Semi-Auto	10	Petrol	145	32.5	2.0	VW
8	8	T-Roc	2019	21995	Manual	10	Petrol	145	44.1	1.0	VW
9	9	T-Roc	2019	23285	Manual	10	Petrol	145	42.2	1.5	VW

In [13]: df['mileage'].value_counts()

Out[13]:

10	5
13904	1
4562	1
6500	1
7414	1
4825	1

Name: mileage, dtype: int64

```
In [22]: x=df[['year']]
         y=df['mileage']
```

```
In [23]: g1={"g":{"g":1,'g':2}}
         df=df.replace(g1)
         print(df)
```

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg
\									
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8
5	5	T-Roc	2020	31895	Manual	10	Petrol	145	42.2
6	6	T-Roc	2020	27895	Manual	10	Petrol	145	42.2
7	7	T-Roc	2020	39495	Semi-Auto	10	Petrol	145	32.5
8	8	T-Roc	2019	21995	Manual	10	Petrol	145	44.1
9	9	T-Roc	2019	23285	Manual	10	Petrol	145	42.2

	engineSize	Make
0	2.0	VW
1	2.0	VW
2	2.0	VW
3	2.0	VW
4	1.5	VW
5	1.5	VW
6	1.5	VW
7	2.0	VW
8	1.0	VW
9	1.5	VW

```
In [24]: 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 [25]: from sklearn.ensemble import RandomForestClassifier

         rfc = RandomForestClassifier()
         rfc.fit(x_train,y_train)
```

```
Out[25]: RandomForestClassifier()
```

```
In [26]: parameters = { 'max_depth':[1,2,3,4,5],
                        'min_samples_leaf':[5,10,15,20,25],
                        'n_estimators':[10,20,30,40,50]
                      }
```

```
In [27]: from sklearn.model_selection import GridSearchCV
```

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:  
666: UserWarning: The least populated class in y has only 1 members, which is  
less than n_splits=2.
```

```
warnings.warn("The least populated class in y has only %d"
```

```
Out[27]: 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')
```

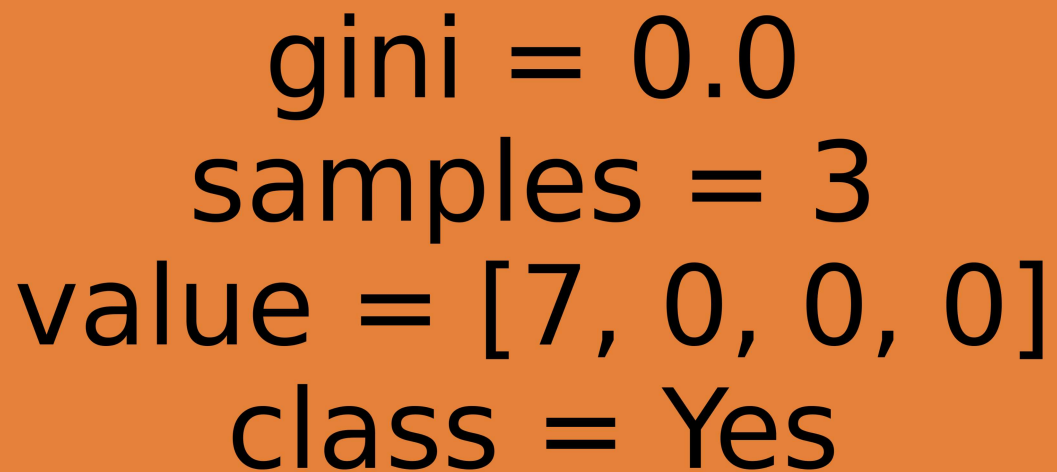
```
In [28]: rf_best=grid_search.best_estimator_  
print(rf_best)
```

```
RandomForestClassifier(max_depth=1, min_samples_leaf=5, n_estimators=10)
```

```
In [29]: from sklearn.tree import plot_tree
```

```
plt.figure(figsize=(80,40))  
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=['Yes','N
```

```
Out[29]: [Text(2232.0, 1087.2, 'gini = 0.0\nsamples = 3\nvalue = [7, 0, 0, 0]\nclass =  
Yes')]
```



gini = 0.0
samples = 3
value = [7, 0, 0, 0]
class = Yes