```python
In [1]: import pandas as pd
        import numpy as np
        from sklearn.preprocessing import StandardScaler
        from sklearn.linear_model import LogisticRegression,LinearRegression
        from sklearn.model_selection import train_test_split
        from numpy import cov
```

```python
In [2]: from sklearn.ensemble import RandomForestClassifier
        import matplotlib.pyplot as plt
        from sklearn.model_selection import GridSearchCV
        from sklearn.tree import plot_tree
```

```python
In [3]: df=pd.read_csv("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

```python
In [4]: df.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 [5]:
```python
y=df["Gender"]
x=df.drop(["Gender"],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [6]:
```python
# lr=LogisticRegression()
# lr.fit(x_train,y_train)
```

In [7]:
```python
# lr.predict(x_test)
```

In [8]:
```python
# lr.score(x_test,y_test)
```

In [9]:
```python
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[9]: RandomForestClassifier()

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

In [11]:
```python
grid_search = GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy
grid_search.fit(x_train,y_train)
```

Out[11]:
```
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')
```
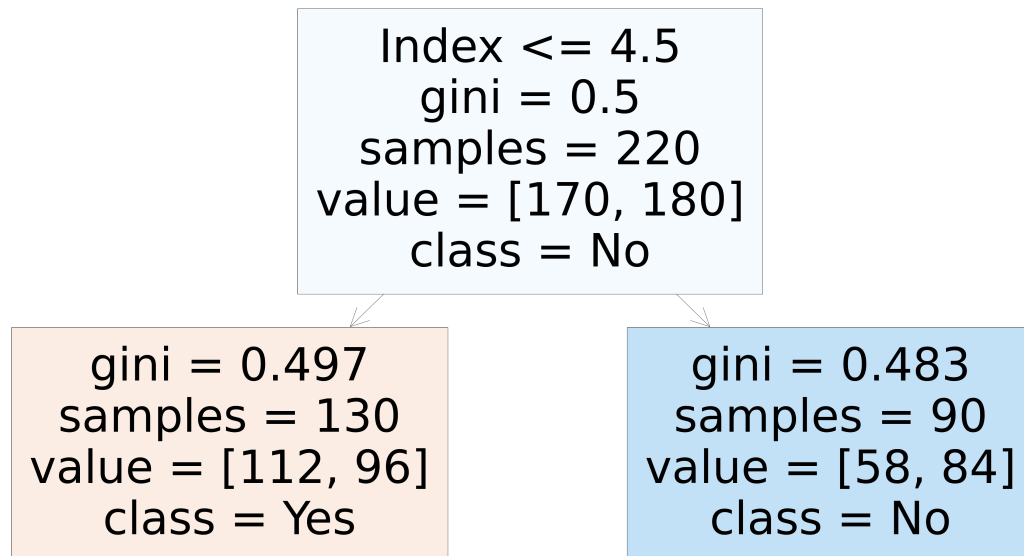
In [12]:
```python
grid_search.best_score_
```

Out[12]: 0.5685714285714285

In [13]:
```python
rfc_best=grid_search.best_estimator_
```

In [14]:
```python
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],
```

Out[14]: [Text(2232.0, 1630.8000000000002, 'Index <= 4.5\ngini = 0.5\nsamples = 220\nvalue = [170, 180]\nclass = No'),
 Text(1116.0, 543.5999999999999, 'gini = 0.497\nsamples = 130\nvalue = [112, 96]\nclass = Yes'),
 Text(3348.0, 543.5999999999999, 'gini = 0.483\nsamples = 90\nvalue = [58, 84]\nclass = No')]

Index <= 4.5
gini = 0.5
samples = 220
value = [170, 180]
class = No

gini = 0.497
samples = 130
value = [112, 96]
class = Yes

gini = 0.483
samples = 90
value = [58, 84]
class = No

In [15]:
```python
df1=pd.read_csv("c7_used_cars.csv")
df1
```

Out[15]:

	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 [16]:
```python
df2=df1.drop(["transmission","Make","model","Unnamed: 0"],axis=1)
df2
```

Out[16]:

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

99187 rows × 7 columns

In [17]:
```python
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99187 entries, 0 to 99186
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   year        99187 non-null  int64
 1   price       99187 non-null  int64
 2   mileage     99187 non-null  int64
 3   fuelType    99187 non-null  object
 4   tax         99187 non-null  int64
 5   mpg         99187 non-null  float64
 6   engineSize  99187 non-null  float64
dtypes: float64(2), int64(4), object(1)
memory usage: 5.3+ MB
```

In [18]:
```python
df2["fuelType"].value_counts()
```

Out[18]:
```
Petrol      54928
Diesel      40928
Hybrid       3078
Other         247
Electric        6
Name: fuelType, dtype: int64
```

In [19]:
```python
f={"fuelType":{"Petrol":1,"Diesel":2,"Hybrid":3,"Other":4,"Electric":5}}
df2=df2.replace(f)
```

In [20]: `df2`

Out[20]:

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

99187 rows × 7 columns

In [21]:
```python
y=df2["fuelType"]
x=df2.drop(["fuelType"],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [22]:
```python
# lr=LogisticRegression(max_iter=10000)
# lr.fit(x_train,y_train)
```

In [23]:
```python
# val=[[2019,25000,16545,145,44.6,1],[2018,68748,1235,108,38,2]]
# lr.predict(val)
```

In [24]:
```python
# lr.score(x_test,y_test)
```

In [25]:
```python
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[25]: `RandomForestClassifier()`

In [26]:
```python
grid_search = GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy
grid_search.fit(x_train,y_train)
```

Out[26]:
```
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')
```
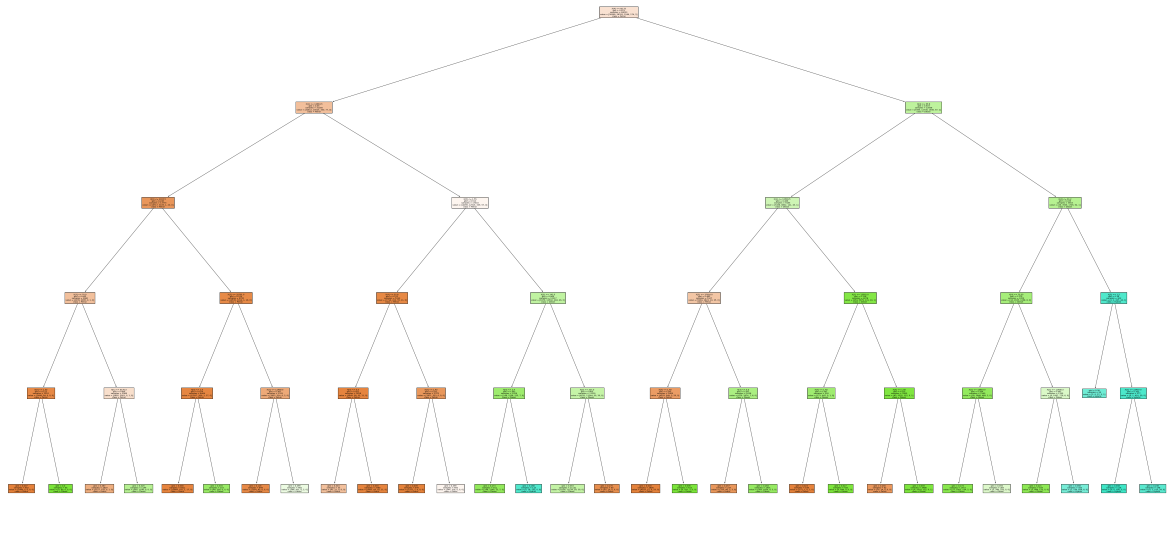
In [27]: `grid_search.best_score_`

Out[27]: `0.9188535215324788`

In [28]: ```python
rfc_best=grid_search.best_estimator_
```

In [29]: ```python
class_name=["Petrol","Diesel","Hybrid","Other","Electric"]
```

In [30]: ```python
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],class_names=class_name,filled=True)
```

```
10, 34, 0]\nclass = Hybrid')]
```


In [31]: ```python
df3=pd.read_csv("C8_loan-train.csv")
df4=pd.read_csv("C8_loan-test.csv")
df3
```

Out[31]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplicant
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [32]:
```python
df3["Loan_Status"]=df3["Loan_Status"].replace("Y",1,regex=True)
df3["Loan_Status"]=df3["Loan_Status"].replace("N",0,regex=True)
df3
```

Out[32]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplicant
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [33]:
```python
df3_tr=df3.drop(["Dependents","Married","Loan_ID","Education","Gender","Property_Ar
df3_tr
```

Out[33]:

	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_His
0	No	5849	0.0	NaN	360.0	
1	No	4583	1508.0	128.0	360.0	
2	Yes	3000	0.0	66.0	360.0	
3	No	2583	2358.0	120.0	360.0	
4	No	6000	0.0	141.0	360.0	
...	...	...	...	...	...	
609	No	2900	0.0	71.0	360.0	
610	No	4106	0.0	40.0	180.0	
611	No	8072	240.0	253.0	360.0	
612	No	7583	0.0	187.0	360.0	
613	Yes	4583	0.0	133.0	360.0	

614 rows × 6 columns

In [34]:
```python
df_tr=df3_tr.dropna()
```

In [35]: `df_tr.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 504 entries, 1 to 613
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Self_Employed     504 non-null    object
 1   ApplicantIncome   504 non-null    int64
 2   CoapplicantIncome 504 non-null    float64
 3   LoanAmount        504 non-null    float64
 4   Loan_Amount_Term  504 non-null    float64
 5   Credit_History    504 non-null    float64
dtypes: float64(4), int64(1), object(1)
memory usage: 27.6+ KB
```

In [36]:
```python
g1={"Self_Employed":{'Yes':1,'No':0}}
df_tr=df_tr.replace(g1).astype(int)
```

In [37]: `df_tr.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 504 entries, 1 to 613
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Self_Employed     504 non-null    int32
 1   ApplicantIncome   504 non-null    int32
 2   CoapplicantIncome 504 non-null    int32
 3   LoanAmount        504 non-null    int32
 4   Loan_Amount_Term  504 non-null    int32
 5   Credit_History    504 non-null    int32
dtypes: int32(6)
memory usage: 15.8 KB
```

In [38]:
```python
y=df_tr["Self_Employed"]
x=df_tr.drop(["Self_Employed"],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
# f=StandardScaler().fit_transform(x)
# lr.fit(f,y)
```

In [39]:
```python
# df4_te=df4.drop(["Education","Loan_ID","Gender","Married","Dependents","Property_
```

In [40]:
```python
# df4_te=df4_te.dropna()
```

In [41]:
```python
# df4_te.info()
```

In [42]: `df_tr["Self_Employed"].value_counts()`

Out[42]:
```
0    434
1     70
Name: Self_Employed, dtype: int64
```

```
In [43]:  # lr.predict(df4_te)
```

```
In [44]:  # lr.predict_proba(df4_te)
```

```
In [45]:  parameter={'max_depth':[1,2,3,4,5],
                     "min_samples_leaf":[5,10,15,20,25],
                     "n_estimators":[10,20,30,40,50]}
```

```
In [46]:  rfc=RandomForestClassifier()
          rfc.fit(x_train,y_train)
```

```
Out[46]:  RandomForestClassifier()
```

```
In [47]:  grid_search = GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy
          grid_search.fit(x_train,y_train)
```

```
Out[47]:  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 [48]:  grid_search.best_score_
```

```
Out[48]:  0.8579545454545454
```

In [49]:
```python
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],class_names=['Yes','No',"Yes"],filled=True)
```

Out[49]: [Text(2352.9, 1993.2, 'X[4] <= 60.75\ngini = 0.523\nsamples = 44023\nvalue = [3830
6, 28762, 2186, 174, 2]\nclass = Yes'),
 Text(1190.4, 1630.8000000000002, 'X[1] <= 14684.5\ngini = 0.45\nsamples = 31185\n
value = [32812, 16024, 340, 77, 0]\nclass = Yes'),
 Text(595.2, 1268.4, 'X[0] <= 2015.5\ngini = 0.246\nsamples = 13775\nvalue = [1871
7, 3109, 1, 20, 0]\nclass = Yes'),
 Text(297.6, 906.0, 'X[3] <= 70.0\ngini = 0.445\nsamples = 3980\nvalue = [4178, 20
95, 0, 1, 0]\nclass = Yes'),
 Text(148.8, 543.5999999999999, 'X[5] <= 1.55\ngini = 0.112\nsamples = 1012\nvalue
= [1486, 94, 0, 0, 0]\nclass = Yes'),
 Text(74.4, 181.19999999999982, 'gini = 0.007\nsamples = 952\nvalue = [1486, 5, 0,
0, 0]\nclass = Yes'),
 Text(223.20000000000002, 181.19999999999982, 'gini = 0.0\nsamples = 60\nvalue =
[0, 89, 0, 0, 0]\nclass = No'),
 Text(446.40000000000003, 543.5999999999999, 'X[1] <= 9170.5\ngini = 0.489\nsample
s = 2968\nvalue = [2692, 2001, 0, 1, 0]\nclass = Yes'),
 Text(372.0, 181.19999999999982, 'gini = 0.387\nsamples = 1824\nvalue = [2127, 75
5, 0, 1, 0]\nclass = Yes'),
 Text(520.8000000000001, 181.19999999999982, 'gini = 0.429\nsamples = 1144\nvalue
= [565, 1246, 0, 0, 0]\nclass = No'),
 Text(892.8000000000001, 906.0, 'X[2] <= 32791.0\ngini = 0.124\nsamples = 9795\nva
lue = [14539, 1014, 1, 19, 0]\nclass = Yes'),
 Text(744.0, 543.5999999999999, 'X[5] <= 1.9\ngini = 0.074\nsamples = 8464\nvalue
= [12917, 500, 1, 17, 0]\nclass = Yes'),
 Text(669.6, 181.19999999999982, 'gini = 0.046\nsamples = 8311\nvalue = [12869, 29
1, 1, 17, 0]\nclass = Yes'),
 Text(818.4000000000001, 181.19999999999982, 'gini = 0.304\nsamples = 153\nvalue =
[48, 209, 0, 0, 0]\nclass = No'),
 Text(1041.6000000000001, 543.5999999999999, 'X[1] <= 11499.5\ngini = 0.367\nsampl
es = 1331\nvalue = [1622, 514, 0, 2, 0]\nclass = Yes'),
 Text(967.2, 181.19999999999982, 'gini = 0.116\nsamples = 849\nvalue = [1254, 83,
0, 0, 0]\nclass = Yes'),
 Text(1116.0, 181.19999999999982, 'gini = 0.499\nsamples = 482\nvalue = [368, 431,
0, 2, 0]\nclass = No'),
 Text(1785.6000000000001, 1268.4, 'X[5] <= 1.55\ngini = 0.513\nsamples = 17410\nva
lue = [14095, 12915, 339, 57, 0]\nclass = Yes'),
 Text(1488.0, 906.0, 'X[4] <= 52.8\ngini = 0.111\nsamples = 5193\nvalue = [7716, 4
13, 55, 11, 0]\nclass = Yes'),
 Text(1339.2, 543.5999999999999, 'X[5] <= 0.5\ngini = 0.051\nsamples = 3818\nvalue
= [5845, 90, 55, 11, 0]\nclass = Yes'),
 Text(1264.8000000000002, 181.19999999999982, 'gini = 0.444\nsamples = 33\nvalue =
[42, 21, 0, 0, 0]\nclass = Yes'),
 Text(1413.6000000000001, 181.19999999999982, 'gini = 0.045\nsamples = 3785\nvalue
= [5803, 69, 55, 11, 0]\nclass = Yes'),
 Text(1636.8000000000002, 543.5999999999999, 'X[5] <= 1.45\ngini = 0.251\nsamples
= 1375\nvalue = [1871, 323, 0, 0, 0]\nclass = Yes'),
 Text(1562.4, 181.19999999999982, 'gini = 0.009\nsamples = 970\nvalue = [1527, 7,
0, 0, 0]\nclass = Yes'),
 Text(1711.2, 181.19999999999982, 'gini = 0.499\nsamples = 405\nvalue = [344, 316,
0, 0, 0]\nclass = Yes'),
 Text(2083.200000000003, 906.0, 'X[3] <= 142.5\ngini = 0.466\nsamples = 12217\nva
lue = [6379, 12502, 284, 46, 0]\nclass = No'),
 Text(1934.4, 543.5999999999999, 'X[5] <= 2.4\ngini = 0.383\nsamples = 1026\nvalue
= [148, 1258, 225, 7, 0]\nclass = No'),
 Text(1860.0000000000002, 181.19999999999982, 'gini = 0.273\nsamples = 911\nvalue
= [148, 1240, 72, 7, 0]\nclass = No'),
 Text(2008.8000000000002, 181.19999999999982, 'gini = 0.188\nsamples = 115\nvalue
= [0, 18, 153, 0, 0]\nclass = Yes'),
 Text(2232.0, 543.5999999999999, 'X[3] <= 327.5\ngini = 0.465\nsamples = 11191\nva
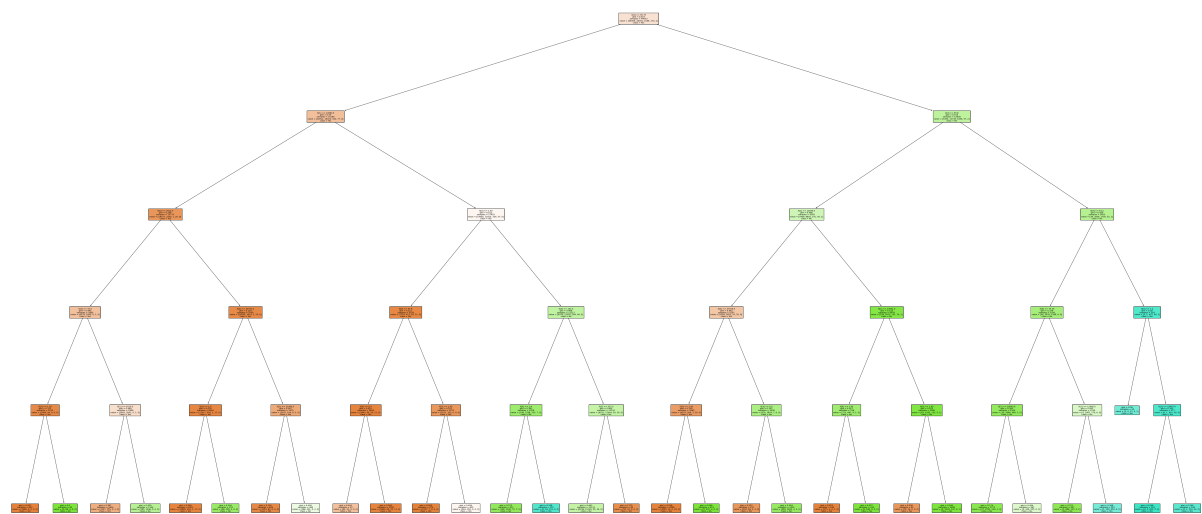lue = [6231, 11244, 59, 39, 0]\nclass = No'),

```
 Text(2157.6000000000004, 181.19999999999982, 'gini = 0.463\nsamples = 11131\nvalu
e = [6138, 11234, 59, 39, 0]\nclass = No'),
 Text(2306.4, 181.19999999999982, 'gini = 0.175\nsamples = 60\nvalue = [93, 10, 0,
0, 0]\nclass = Yes'),
 Text(3515.4, 1630.8000000000002, 'X[4] <= 69.8\ngini = 0.519\nsamples = 12838\nva
lue = [5494, 12738, 1846, 97, 2]\nclass = No'),
 Text(2976.0, 1268.4, 'X[1] <= 12599.5\ngini = 0.486\nsamples = 9185\nvalue = [545
8, 8802, 141, 45, 1]\nclass = No'),
 Text(2678.4, 906.0, 'X[2] <= 37016.0\ngini = 0.462\nsamples = 5107\nvalue = [520
4, 2823, 10, 35, 0]\nclass = Yes'),
 Text(2529.6000000000004, 543.5999999999999, 'X[5] <= 1.45\ngini = 0.295\nsamples
= 3469\nvalue = [4525, 949, 3, 29, 0]\nclass = Yes'),
 Text(2455.2000000000003, 181.19999999999982, 'gini = 0.018\nsamples = 2856\nvalue
= [4525, 12, 0, 29, 0]\nclass = Yes'),
 Text(2604.0, 181.19999999999982, 'gini = 0.006\nsamples = 613\nvalue = [0, 937,
3, 0, 0]\nclass = No'),
 Text(2827.2000000000003, 543.5999999999999, 'X[3] <= 5.0\ngini = 0.397\nsamples =
1638\nvalue = [679, 1874, 7, 6, 0]\nclass = No'),
 Text(2752.8, 181.19999999999982, 'gini = 0.257\nsamples = 237\nvalue = [310, 48,
3, 3, 0]\nclass = Yes'),
 Text(2901.6000000000004, 181.19999999999982, 'gini = 0.284\nsamples = 1401\nvalue
= [369, 1826, 4, 3, 0]\nclass = No'),
 Text(3273.6000000000004, 906.0, 'X[1] <= 13501.0\ngini = 0.118\nsamples = 4078\nv
alue = [254, 5979, 131, 10, 1]\nclass = No'),
 Text(3124.8, 543.5999999999999, 'X[5] <= 1.45\ngini = 0.347\nsamples = 518\nvalue
= [171, 647, 9, 1, 0]\nclass = No'),
 Text(3050.4, 181.19999999999982, 'gini = 0.023\nsamples = 108\nvalue = [171, 1,
0, 1, 0]\nclass = Yes'),
 Text(3199.2000000000003, 181.19999999999982, 'gini = 0.027\nsamples = 410\nvalue
= [0, 646, 9, 0, 0]\nclass = No'),
 Text(3422.4, 543.5999999999999, 'X[5] <= 1.45\ngini = 0.075\nsamples = 3560\nvalu
e = [83, 5332, 122, 9, 1]\nclass = No'),
 Text(3348.0000000000005, 181.19999999999982, 'gini = 0.256\nsamples = 62\nvalue =
[79, 14, 0, 0, 0]\nclass = Yes'),
 Text(3496.8, 181.19999999999982, 'gini = 0.049\nsamples = 3498\nvalue = [4, 5318,
122, 9, 1]\nclass = No'),
 Text(4054.8, 1268.4, 'X[4] <= 91.2\ngini = 0.439\nsamples = 3653\nvalue = [36, 39
36, 1705, 52, 1]\nclass = No'),
 Text(3868.8, 906.0, 'X[4] <= 75.35\ngini = 0.379\nsamples = 3350\nvalue = [34, 39
35, 1268, 8, 0]\nclass = No'),
 Text(3720.0000000000005, 543.5999999999999, 'X[1] <= 18993.0\ngini = 0.264\nsampl
es = 2160\nvalue = [30, 2848, 489, 2, 0]\nclass = No'),
 Text(3645.6000000000004, 181.19999999999982, 'gini = 0.174\nsamples = 1773\nvalue
= [27, 2502, 234, 2, 0]\nclass = No'),
 Text(3794.4, 181.19999999999982, 'gini = 0.494\nsamples = 387\nvalue = [3, 346, 2
55, 0, 0]\nclass = No'),
 Text(4017.6000000000004, 543.5999999999999, 'X[1] <= 11400.5\ngini = 0.492\nsampl
es = 1190\nvalue = [4, 1087, 779, 6, 0]\nclass = No'),
 Text(3943.2000000000003, 181.19999999999982, 'gini = 0.221\nsamples = 637\nvalue
= [3, 888, 125, 0, 0]\nclass = No'),
 Text(4092.0000000000005, 181.19999999999982, 'gini = 0.368\nsamples = 553\nvalue
= [1, 199, 654, 6, 0]\nclass = Yes'),
 Text(4240.8, 906.0, 'X[5] <= 1.2\ngini = 0.18\nsamples = 303\nvalue = [2, 1, 437,
44, 1]\nclass = Yes'),
 Text(4166.400000000001, 543.5999999999999, 'gini = 0.38\nsamples = 26\nvalue =
[2, 0, 27, 5, 1]\nclass = Yes'),
 Text(4315.200000000001, 543.5999999999999, 'X[2] <= 23497.0\ngini = 0.162\nsample
s = 277\nvalue = [0, 1, 410, 39, 0]\nclass = Yes'),
 Text(4240.8, 181.19999999999982, 'gini = 0.048\nsamples = 129\nvalue = [0, 0, 20
0, 5, 0]\nclass = Yes'),
```

```
 Text(4389.6, 181.19999999999982, 'gini = 0.246\nsamples = 148\nvalue = [0, 1, 21
 0, 34, 0]\nclass = Yes')]
```


In [50]:
```
df5=pd.read_csv("C9_Data.csv")
df5
```

Out[50]:

	row_id	user_id	timestamp	gate_id
0	0	18	2022-07-29 09:08:54	7
1	1	18	2022-07-29 09:09:54	9
2	2	18	2022-07-29 09:09:54	9
3	3	18	2022-07-29 09:10:06	5
4	4	18	2022-07-29 09:10:08	5
...	...	...	...	...
37513	37513	6	2022-12-31 20:38:56	11
37514	37514	6	2022-12-31 20:39:22	6
37515	37515	6	2022-12-31 20:39:23	6
37516	37516	6	2022-12-31 20:39:31	9
37517	37517	6	2022-12-31 20:39:31	9

37518 rows × 4 columns

In [51]:
```python
df5=df5.drop(["timestamp"],axis=1)
df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37518 entries, 0 to 37517
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   row_id    37518 non-null  int64
 1   user_id   37518 non-null  int64
 2   gate_id   37518 non-null  int64
dtypes: int64(3)
memory usage: 879.5 KB
```

In [52]:
```python
y=df5["user_id"]
x=df5.drop(["user_id"],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [53]:
```python
# lr=LogisticRegression(max_iter=10000)
# li=LinearRegression()
# lr.fit(x_train,y_train)
# li.fit(x_train,y_train)
```

In [54]:
```python
# lr.predict(x_test)
```

In [55]:
```python
# li.predict(x_test)
```

In [56]:
```python
# lr.score(x_test,y_test)
```

In [57]:
```python
# li.score(x_test,y_test)
```

In [58]:
```python
# cov(lr.predict(x_test),li.predict(x_test))
```

In [59]:
```python
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[59]: RandomForestClassifier()

In [60]:
```python
grid_search = GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy
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[60]: 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 [61]: `grid_search.best_score_`

Out[61]: 0.1067702383672226

In [62]:
```python
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],class_names=['Yes','No','Yes'],filled=True)
```

Out[62]: [Text(2352.9, 1993.2, 'X[4] <= 60.75\ngini = 0.523\nsamples = 44023\nvalue = [3830
          6, 28762, 2186, 174, 2]\nclass = Yes'),
           Text(1190.4, 1630.8000000000002, 'X[1] <= 14684.5\ngini = 0.45\nsamples = 31185\n
          value = [32812, 16024, 340, 77, 0]\nclass = Yes'),
           Text(595.2, 1268.4, 'X[0] <= 2015.5\ngini = 0.246\nsamples = 13775\nvalue = [1871
          7, 3109, 1, 20, 0]\nclass = Yes'),
           Text(297.6, 906.0, 'X[3] <= 70.0\ngini = 0.445\nsamples = 3980\nvalue = [4178, 20
          95, 0, 1, 0]\nclass = Yes'),
           Text(148.8, 543.5999999999999, 'X[5] <= 1.55\ngini = 0.112\nsamples = 1012\nvalue
          = [1486, 94, 0, 0, 0]\nclass = Yes'),
           Text(74.4, 181.19999999999982, 'gini = 0.007\nsamples = 952\nvalue = [1486, 5, 0,
          0, 0]\nclass = Yes'),
           Text(223.20000000000002, 181.19999999999982, 'gini = 0.0\nsamples = 60\nvalue =
          [0, 89, 0, 0, 0]\nclass = No'),
           Text(446.40000000000003, 543.5999999999999, 'X[1] <= 9170.5\ngini = 0.489\nsample
          s = 2968\nvalue = [2692, 2001, 0, 1, 0]\nclass = Yes'),
           Text(372.0, 181.19999999999982, 'gini = 0.387\nsamples = 1824\nvalue = [2127, 75
          5, 0, 1, 0]\nclass = Yes'),
           Text(520.8000000000001, 181.19999999999982, 'gini = 0.429\nsamples = 1144\nvalue
          = [565, 1246, 0, 0, 0]\nclass = No'),
           Text(892.8000000000001, 906.0, 'X[2] <= 32791.0\ngini = 0.124\nsamples = 9795\nva
          lue = [14539, 1014, 1, 19, 0]\nclass = Yes'),
           Text(744.0, 543.5999999999999, 'X[5] <= 1.9\ngini = 0.074\nsamples = 8464\nvalue
          = [12917, 500, 1, 17, 0]\nclass = Yes'),
           Text(669.6, 181.19999999999982, 'gini = 0.046\nsamples = 8311\nvalue = [12869, 29
          1, 1, 17, 0]\nclass = Yes'),
           Text(818.4000000000001, 181.19999999999982, 'gini = 0.304\nsamples = 153\nvalue =
          [48, 209, 0, 0, 0]\nclass = No'),
           Text(1041.6000000000001, 543.5999999999999, 'X[1] <= 11499.5\ngini = 0.367\nsampl
          es = 1331\nvalue = [1622, 514, 0, 2, 0]\nclass = Yes'),
           Text(967.2, 181.19999999999982, 'gini = 0.116\nsamples = 849\nvalue = [1254, 83,
          0, 0, 0]\nclass = Yes'),
           Text(1116.0, 181.19999999999982, 'gini = 0.499\nsamples = 482\nvalue = [368, 431,
          0, 2, 0]\nclass = No'),
           Text(1785.6000000000001, 1268.4, 'X[5] <= 1.55\ngini = 0.513\nsamples = 17410\nva
          lue = [14095, 12915, 339, 57, 0]\nclass = Yes'),
           Text(1488.0, 906.0, 'X[4] <= 52.8\ngini = 0.111\nsamples = 5193\nvalue = [7716, 4
          13, 55, 11, 0]\nclass = Yes'),
           Text(1339.2, 543.5999999999999, 'X[5] <= 0.5\ngini = 0.051\nsamples = 3818\nvalue
          = [5845, 90, 55, 11, 0]\nclass = Yes'),
           Text(1264.8000000000002, 181.19999999999982, 'gini = 0.444\nsamples = 33\nvalue =
          [42, 21, 0, 0, 0]\nclass = Yes'),
           Text(1413.6000000000001, 181.19999999999982, 'gini = 0.045\nsamples = 3785\nvalue
          = [5803, 69, 55, 11, 0]\nclass = Yes'),
           Text(1636.8000000000002, 543.5999999999999, 'X[5] <= 1.45\ngini = 0.251\nsamples
          = 1375\nvalue = [1871, 323, 0, 0, 0]\nclass = Yes'),
           Text(1562.4, 181.19999999999982, 'gini = 0.009\nsamples = 970\nvalue = [1527, 7,
          0, 0, 0]\nclass = Yes'),
           Text(1711.2, 181.19999999999982, 'gini = 0.499\nsamples = 405\nvalue = [344, 316,
          0, 0, 0]\nclass = Yes'),
           Text(2083.2000000000003, 906.0, 'X[3] <= 142.5\ngini = 0.466\nsamples = 12217\nva
          lue = [6379, 12502, 284, 46, 0]\nclass = No'),
           Text(1934.4, 543.5999999999999, 'X[5] <= 2.4\ngini = 0.383\nsamples = 1026\nvalue
          = [148, 1258, 225, 7, 0]\nclass = No'),
           Text(1860.0000000000002, 181.19999999999982, 'gini = 0.273\nsamples = 911\nvalue
          = [148, 1240, 72, 7, 0]\nclass = No'),
           Text(2008.8000000000002, 181.19999999999982, 'gini = 0.188\nsamples = 115\nvalue
          = [0, 18, 153, 0, 0]\nclass = Yes'),
           Text(2232.0, 543.5999999999999, 'X[3] <= 327.5\ngini = 0.465\nsamples = 11191\nva
          lue = [6231, 11244, 59, 39, 0]\nclass = No'),
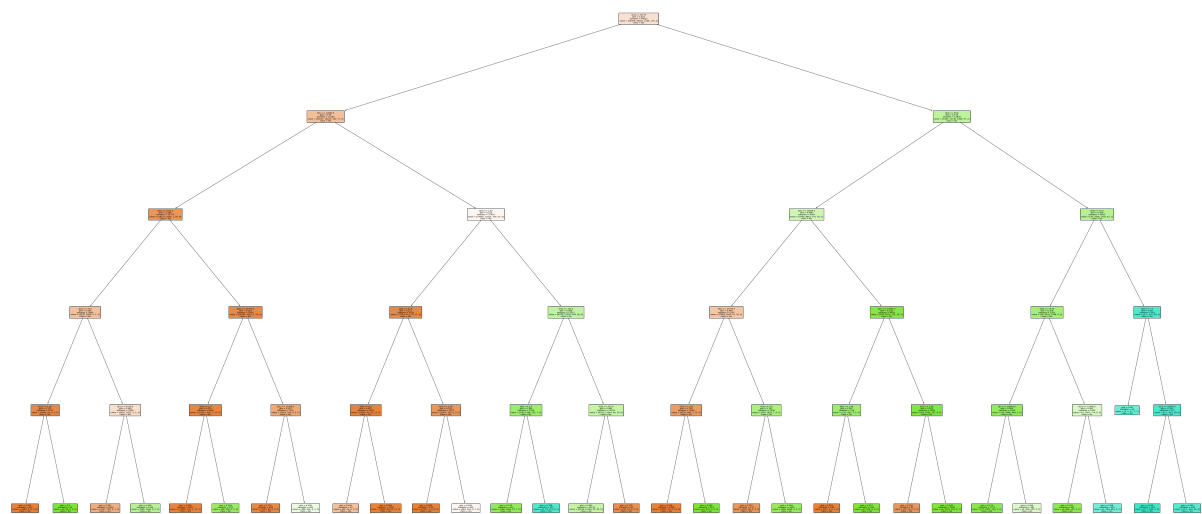
```
      Text(2157.6000000000004, 181.19999999999982, 'gini = 0.463\nsamples = 11131\nvalu
e = [6138, 11234, 59, 39, 0]\nclass = No'),
      Text(2306.4, 181.19999999999982, 'gini = 0.175\nsamples = 60\nvalue = [93, 10, 0,
0, 0]\nclass = Yes'),
      Text(3515.4, 1630.8000000000002, 'X[4] <= 69.8\ngini = 0.519\nsamples = 12838\nva
lue = [5494, 12738, 1846, 97, 2]\nclass = No'),
      Text(2976.0, 1268.4, 'X[1] <= 12599.5\ngini = 0.486\nsamples = 9185\nvalue = [545
8, 8802, 141, 45, 1]\nclass = No'),
      Text(2678.4, 906.0, 'X[2] <= 37016.0\ngini = 0.462\nsamples = 5107\nvalue = [520
4, 2823, 10, 35, 0]\nclass = Yes'),
      Text(2529.6000000000004, 543.5999999999999, 'X[5] <= 1.45\ngini = 0.295\nsamples
= 3469\nvalue = [4525, 949, 3, 29, 0]\nclass = Yes'),
      Text(2455.2000000000003, 181.19999999999982, 'gini = 0.018\nsamples = 2856\nvalue
= [4525, 12, 0, 29, 0]\nclass = Yes'),
      Text(2604.0, 181.19999999999982, 'gini = 0.006\nsamples = 613\nvalue = [0, 937,
3, 0, 0]\nclass = No'),
      Text(2827.2000000000003, 543.5999999999999, 'X[3] <= 5.0\ngini = 0.397\nsamples =
1638\nvalue = [679, 1874, 7, 6, 0]\nclass = No'),
      Text(2752.8, 181.19999999999982, 'gini = 0.257\nsamples = 237\nvalue = [310, 48,
3, 3, 0]\nclass = Yes'),
      Text(2901.6000000000004, 181.19999999999982, 'gini = 0.284\nsamples = 1401\nvalue
= [369, 1826, 4, 3, 0]\nclass = No'),
      Text(3273.6000000000004, 906.0, 'X[1] <= 13501.0\ngini = 0.118\nsamples = 4078\nv
alue = [254, 5979, 131, 10, 1]\nclass = No'),
      Text(3124.8, 543.5999999999999, 'X[5] <= 1.45\ngini = 0.347\nsamples = 518\nvalue
= [171, 647, 9, 1, 0]\nclass = No'),
      Text(3050.4, 181.19999999999982, 'gini = 0.023\nsamples = 108\nvalue = [171, 1,
0, 1, 0]\nclass = Yes'),
      Text(3199.2000000000003, 181.19999999999982, 'gini = 0.027\nsamples = 410\nvalue
= [0, 646, 9, 0, 0]\nclass = No'),
      Text(3422.4, 543.5999999999999, 'X[5] <= 1.45\ngini = 0.075\nsamples = 3560\nvalu
e = [83, 5332, 122, 9, 1]\nclass = No'),
      Text(3348.0000000000005, 181.19999999999982, 'gini = 0.256\nsamples = 62\nvalue =
[79, 14, 0, 0, 0]\nclass = Yes'),
      Text(3496.8, 181.19999999999982, 'gini = 0.049\nsamples = 3498\nvalue = [4, 5318,
122, 9, 1]\nclass = No'),
      Text(4054.8, 1268.4, 'X[4] <= 91.2\ngini = 0.439\nsamples = 3653\nvalue = [36, 39
36, 1705, 52, 1]\nclass = No'),
      Text(3868.8, 906.0, 'X[4] <= 75.35\ngini = 0.379\nsamples = 3350\nvalue = [34, 39
35, 1268, 8, 0]\nclass = No'),
      Text(3720.0000000000005, 543.5999999999999, 'X[1] <= 18993.0\ngini = 0.264\nsampl
es = 2160\nvalue = [30, 2848, 489, 2, 0]\nclass = No'),
      Text(3645.6000000000004, 181.19999999999982, 'gini = 0.174\nsamples = 1773\nvalue
= [27, 2502, 234, 2, 0]\nclass = No'),
      Text(3794.4, 181.19999999999982, 'gini = 0.494\nsamples = 387\nvalue = [3, 346, 2
55, 0, 0]\nclass = No'),
      Text(4017.6000000000004, 543.5999999999999, 'X[1] <= 11400.5\ngini = 0.492\nsampl
es = 1190\nvalue = [4, 1087, 779, 6, 0]\nclass = No'),
      Text(3943.2000000000003, 181.19999999999982, 'gini = 0.221\nsamples = 637\nvalue
= [3, 888, 125, 0, 0]\nclass = No'),
      Text(4092.0000000000005, 181.19999999999982, 'gini = 0.368\nsamples = 553\nvalue
= [1, 199, 654, 6, 0]\nclass = Yes'),
      Text(4240.8, 906.0, 'X[5] <= 1.2\ngini = 0.18\nsamples = 303\nvalue = [2, 1, 437,
44, 1]\nclass = Yes'),
      Text(4166.400000000001, 543.5999999999999, 'gini = 0.38\nsamples = 26\nvalue =
[2, 0, 27, 5, 1]\nclass = Yes'),
      Text(4315.200000000001, 543.5999999999999, 'X[2] <= 23497.0\ngini = 0.162\nsample
s = 277\nvalue = [0, 1, 410, 39, 0]\nclass = Yes'),
      Text(4240.8, 181.19999999999982, 'gini = 0.048\nsamples = 129\nvalue = [0, 0, 20
0, 5, 0]\nclass = Yes'),
```

```
    Text(4389.6, 181.19999999999982, 'gini = 0.246\nsamples = 148\nvalue = [0, 1, 21
    0, 34, 0]\nclass = Yes')]
```


In [ ]:

In [ ]: