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\C10_loan1 - C10_loan1.csv")
df
```

Out[3]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	Yes	Single	125	No
1	No	Married	100	No
2	No	Single	70	No
3	Yes	Married	120	No
4	No	Divorced	95	Yes
5	No	Married	60	No
6	Yes	Divorced	220	No
7	No	Single	85	Yes
8	No	Married	75	No
9	No	Single	90	Yes

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Home Owner          10 non-null     object
 1   Marital Status      10 non-null     object
 2   Annual Income       10 non-null     int64 
 3   Defaulted Borrower  10 non-null     object
dtypes: int64(1), object(3)
memory usage: 448.0+ bytes
```

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

Out[5]:
```
Index(['Home Owner', 'Marital Status', 'Annual Income', 'Defaulted Borrower'],
dtype='object')
```

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

Out[6]:
```
No     7
Yes    3
Name: Defaulted Borrower, dtype: int64
```

In [10]:

```
df1=df[['Annual Income','Defaulted Borrower']]
```

In [11]:

```
x=df1[['Annual Income']]
y=df1['Defaulted Borrower']
```

In [12]:

```
g1={'gate_id':{'15':0,'14':1,'13':2,'12':3,'11':4,'10':5,'9':6,'8':7,'7':8,'6':9,
:12,'1':13,'0':14,'-1':15}}
df1=df1.replace(g1)
print(df1)
```

```
   Annual Income Defaulted Borrower
0            125                 No
1            100                 No
2             70                 No
3            120                 No
4             95                Yes
5             60                 No
6            220                 No
7             85                Yes
8             75                 No
9             90                Yes
```

In [15]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5)
```

In [16]:

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[16]:

```
RandomForestClassifier()
```

In [17]:

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

In [18]:

```
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)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:66
6: UserWarning: The least populated class in y has only 1 members, which is les
s than n_splits=2.
  warnings.warn(("The least populated class in y has only %d"
```

Out[18]:

```
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 [19]:

```
grid_search.best_score_
```

Out[19]:

```
0.8333333333333333
```

In [20]:

```
rfc_best = grid_search.best_estimator_
```

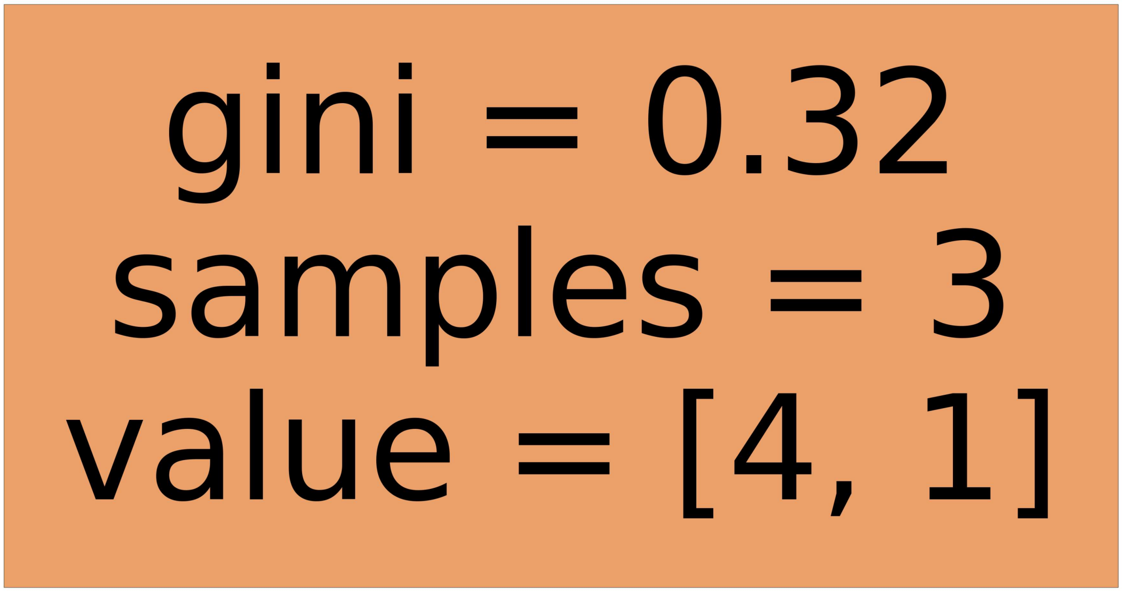
In [21]:

```
from sklearn.tree import plot_tree

plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[4],feature_names=x.columns,filled=True)
```

Out[21]:

```
[Text(2232.0, 1087.2, 'gini = 0.32\nsamples = 3\nvalue = [4, 1]')]
```


gini = 0.32
samples = 3
value = [4, 1]