In [3]:

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns
```

In [10]:

```
df=pd.read_excel('C:/Users/HP/OneDrive/Desktop/DataScience/Customer_Churn.xlsx')
```

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3150 entries, 0 to 3149
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Call  Failure            3150 non-null   int64  
 1   Complains                3150 non-null   int64  
 2   Subscription  Length     3150 non-null   int64  
 3   Charge  Amount           3150 non-null   int64  
 4   Seconds of Use           3150 non-null   int64  
 5   Frequency of use         3150 non-null   int64  
 6   Frequency of SMS         3150 non-null   int64  
 7   Distinct Called Numbers  3150 non-null   int64  
 8   Age Group                3150 non-null   int64  
 9   Tariff Plan              3150 non-null   int64  
 10  Status                   3150 non-null   int64  
 11  Age                      3150 non-null   int64  
 12  Customer Value           3150 non-null   float64
 13  Churn                    3150 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 344.7 KB
```

In [21]:

```
X=df.columns[3:17]
X
```

Out[21]:

```
Index(['Charge  Amount', 'Seconds of Use', 'Frequency of use',
       'Frequency of SMS', 'Distinct Called Numbers', 'Age Group',
       'Tariff Plan', 'Status', 'Age', 'Customer Value', 'Churn'],
      dtype='object')
```

In [18]:

```
encoded_df = pd.get_dummies(df[x],drop_first=True)
list(encoded_df.columns)
```

Out[18]:

```
['Charge  Amount',
 'Seconds of Use',
 'Frequency of use',
 'Frequency of SMS',
 'Distinct Called Numbers',
 'Age Group',
 'Tariff Plan',
 'Status',
 'Age',
 'Customer Value',
 'Churn']
```

In [26]:

```
import statsmodels.api as sn
y = df.Churn
X=sn.add_constant(encoded_df)
```

In [27]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_st
```

In [28]:

```
clf = DecisionTreeClassifier(criterion='gini',max_depth = 3,random_state =20)
```

In [29]:

```
clf = clf.fit(X_train,y_train)
```

In [30]:

```
from sklearn import tree
```

In [31]:

```
classnames = ['Retain','not-retain']
```

In [33]:

```
fig = plt.figure(figsize=(50,40))
```

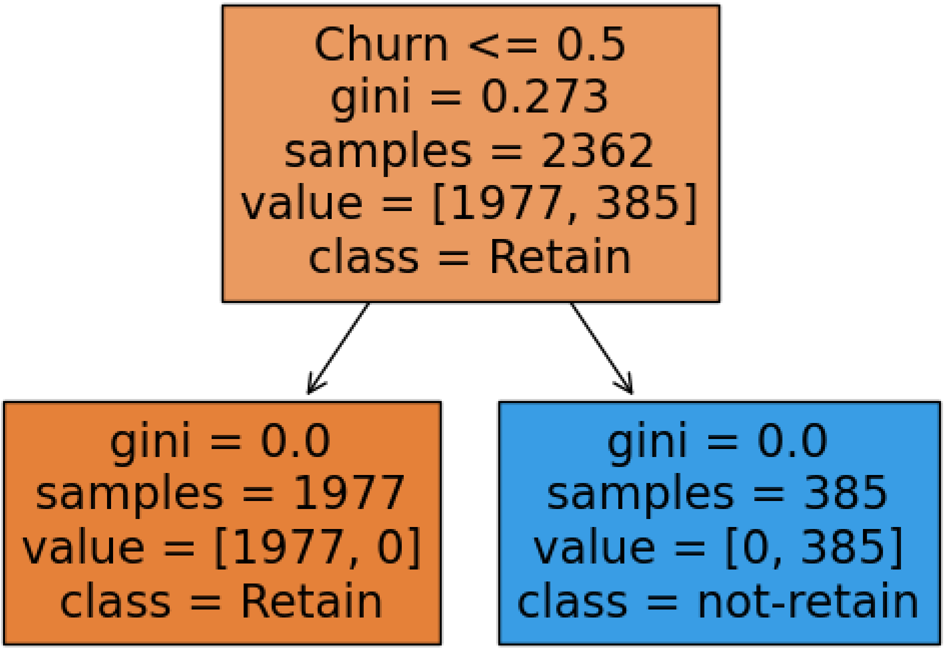
```
<Figure size 5000x4000 with 0 Axes>
```

In [34]:

```
tree.plot_tree(clf,feature_names = X_train.columns,class_names = classnames,filled =
```

Out[34]:

```
[Text(0.5, 0.75, 'Churn <= 0.5\ngini = 0.273\nsamples = 2362\nvalue =
[1977, 385]\nclass = Retain'),
 Text(0.25, 0.25, 'gini = 0.0\nsamples = 1977\nvalue = [1977, 0]\nclass
= Retain'),
 Text(0.75, 0.25, 'gini = 0.0\nsamples = 385\nvalue = [0, 385]\nclass =
not-retain')]
```

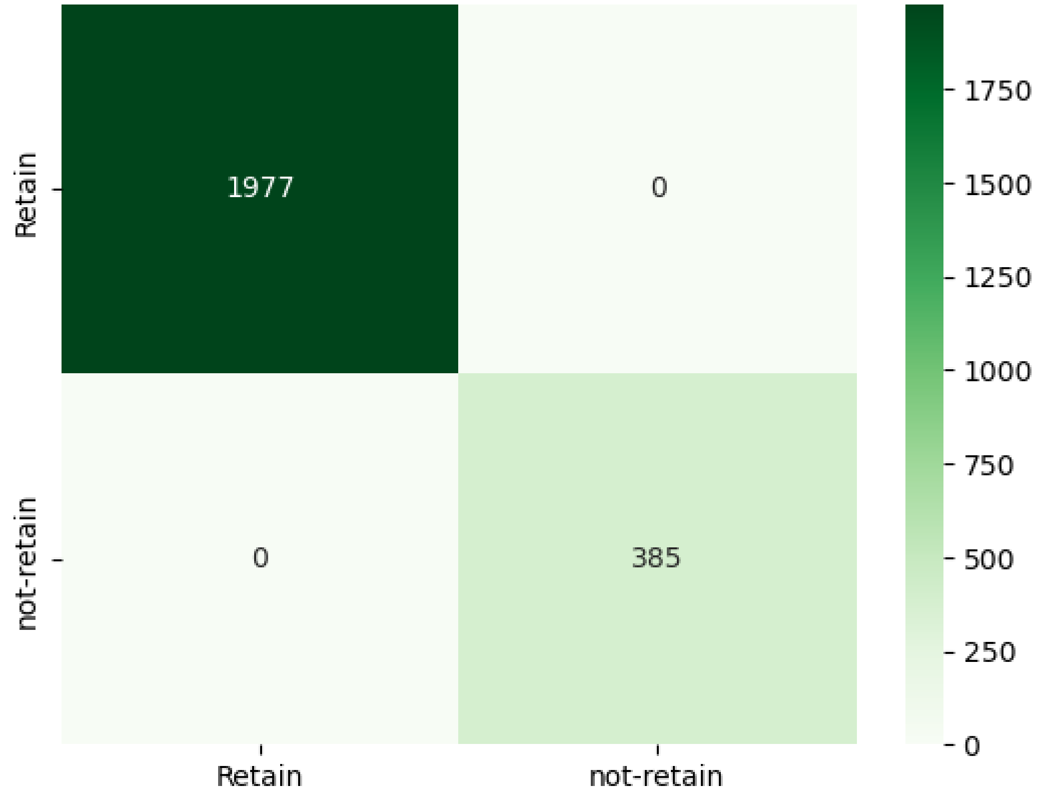


In [43]:

```
from sklearn import metrics
from sklearn.metrics import confusion_matrix
train_tree_predict = clf.predict(X_train)
```

In [44]:

```
#evaluating ROC
classnames = ['Retain','not-retain']
confusionmatrix=confusion_matrix(y_train,train_tree_predict)
sns.heatmap(confusionmatrix,annot=True,yticklabels=classnames,xticklabels=classnames,
```



In [45]:

```
metrics.roc_auc_score(y_train,train_tree_predict)
```

Out[45]:

```
1.0
```

In [1]:

```
#for test data
from sklearn import metrics
from sklearn.metrics import confusion_matrix
test_tree_predict = clf.predict(X_train)
```

```
------------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
Cell In[1], line 4
      2 from sklearn import metrics
      3 from sklearn.metrics import confusion_matrix
----> 4 test_tree_predict = clf.predict(X_train)

NameError: name 'clf' is not defined
```

In [ ]:

```
snames =[]
```