In [1]:
```
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
```

In [2]:
```
s=pd.read_csv("C:/Users/HP/Downloads/winequality_red.csv")
```

In [3]:
```
s.head()
```

Out[3]:

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcohol |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 |
| **1** | 7.8 | 0.88 | 0.00 | 2.6 | 0.098 | 25.0 | 67.0 | 0.9968 | 3.20 | 0.68 | 9.8 |
| **2** | 7.8 | 0.76 | 0.04 | 2.3 | 0.092 | 15.0 | 54.0 | 0.9970 | 3.26 | 0.65 | 9.8 |
| **3** | 11.2 | 0.28 | 0.56 | 1.9 | 0.075 | 17.0 | 60.0 | 0.9980 | 3.16 | 0.58 | 9.8 |
| **4** | 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 |

In [4]:
```
s.shape
```

Out[4]:
```
(1599, 12)
```

In [5]:
```
s.groupby('quality').size()
```

Out[5]:
```
quality
3     10
4     53
5    681
6    638
7    199
8     18
dtype: int64
```

In [6]:
```
_train,X_test,y_train,y_test=train_test_split(s.loc[:,s.columns != 'quality'],s['q
```

In [7]:
```
s.loc[:,s.columns!='quality']
```

Out[7]:

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcoh |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7.4 | 0.700 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.99780 | 3.51 | 0.56 | 9 |
| **1** | 7.8 | 0.880 | 0.00 | 2.6 | 0.098 | 25.0 | 67.0 | 0.99680 | 3.20 | 0.68 | 9 |
| **2** | 7.8 | 0.760 | 0.04 | 2.3 | 0.092 | 15.0 | 54.0 | 0.99700 | 3.26 | 0.65 | 9 |
| **3** | 11.2 | 0.280 | 0.56 | 1.9 | 0.075 | 17.0 | 60.0 | 0.99800 | 3.16 | 0.58 | 9 |
| **4** | 7.4 | 0.700 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.99780 | 3.51 | 0.56 | 9 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **1594** | 6.2 | 0.600 | 0.08 | 2.0 | 0.090 | 32.0 | 44.0 | 0.99490 | 3.45 | 0.58 | 10 |
| **1595** | 5.9 | 0.550 | 0.10 | 2.2 | 0.062 | 39.0 | 51.0 | 0.99512 | 3.52 | 0.76 | 11 |
| **1596** | 6.3 | 0.510 | 0.13 | 2.3 | 0.076 | 29.0 | 40.0 | 0.99574 | 3.42 | 0.75 | 11 |
| **1597** | 5.9 | 0.645 | 0.12 | 2.0 | 0.075 | 32.0 | 44.0 | 0.99547 | 3.57 | 0.71 | 10 |
| **1598** | 6.0 | 0.310 | 0.47 | 3.6 | 0.067 | 18.0 | 42.0 | 0.99549 | 3.39 | 0.66 | 11 |

1599 rows × 11 columns

In [8]:
```
print(y_train.value_counts())
print(y_test.value_counts())
```

```
5    511
6    478
7    149
4     40
8     13
3      8
Name: quality, dtype: int64
5    170
6    160
7     50
4     13
8      5
3      2
Name: quality, dtype: int64
```

In [9]:
```
feature_name=list(X_train.columns)
class_name=list(y_train.unique())
feature_name
```

Out[9]:
```
['fixed acidity',
 'volatile acidity',
 'citric acid',
 'residual sugar',
 'chlorides',
 'free sulfur dioxide',
 'total sulfur dioxide',
 'density',
 'pH',
 'sulphates',
 'alcohol']
```

In [10]:
```
class_name
```

Out[10]:
```
[5, 6, 7, 4, 3, 8]
```

In [11]:
```
A=DecisionTreeClassifier()#create decision tree object
A=A.fit(X_train,y_train)#train DecisionTreeClassifier
y_pred=A.predict(X_test) #predict response for test dataset
```

In [12]:
```
print("Accuracy:",metrics.accuracy_score(y_test,y_pred))
```

```
Accuracy: 0.5925
```

In [13]:
```
from sklearn import tree
plt.figure(figsize=(70,30))
tree.plot_tree(A,filled=True)
```

Out[13]:
```
[Text(1547.123378245142, 1598.184, 'X[10] <= 10.15\ngini = 0.643\nsamples = 1
199\nvalue = [8, 40, 511, 478, 149, 13]'),
 Text(799.8587808190556, 1532.952, 'X[9] <= 0.575\ngini = 0.508\nsamples = 59
9\nvalue = [4, 22, 380, 177, 15, 1]'),
 Text(276.6681989134977, 1467.72, 'X[10] <= 9.075\ngini = 0.384\nsamples = 25
9\nvalue = [2, 13, 198, 44, 2, 0]'),
 Text(180.36481404095278, 1402.4879999999998, 'X[0] <= 8.15\ngini = 0.708\nsa
mples = 12\nvalue = [1, 3, 4, 4, 0, 0]'),
 Text(154.24864187212702, 1337.2559999999999, 'X[1] <= 0.605\ngini = 0.5\nsam
ples = 6\nvalue = [0, 3, 3, 0, 0, 0]'),
 Text(141.19055578771417, 1272.024, 'gini = 0.0\nsamples = 3\nvalue = [0, 3,
0, 0, 0, 0]'),
 Text(167.3067279565399, 1272.024, 'gini = 0.0\nsamples = 3\nvalue = [0, 0,
3, 0, 0, 0]'),
 Text(206.4809862097785, 1337.2559999999999, 'X[2] <= 0.55\ngini = 0.5\nsampl
es = 6\nvalue = [1, 0, 1, 4, 0, 0]'),
 Text(193.42290012536566, 1272.024, 'gini = 0.0\nsamples = 4\nvalue = [0, 0,
0, 4, 0, 0]'),
 Text(219.53907229419139, 1272.024, 'X[9] <= 0.53\ngini = 0.5\nsamples = 2\nv
```

In [ ]:
```

```