In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier,plot_tree
from sklearn.metrics import accuracy_score, classification_report
```

In [5]:
```
df=pd.read_csv('cancer_data.csv')
df.head()
```

Out[5]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothn
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

5 rows × 33 columns

In [7]:
```
y=df[['diagnosis']]
x=df.drop(columns=["diagnosis","id"], axis=1)
```

In [9]:
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.4, random_sta
clf=DecisionTreeClassifier(criterion='entropy', random_state=42)
clf.fit(X_train, y_train)
```

Out[9]:
```
DecisionTreeClassifier(criterion='entropy', random_state=42)
```

In [15]:
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_sta
clf=DecisionTreeClassifier(criterion='entropy', random_state=42)
clf.fit(X_train, y_train)
```

Out[15]:
```
DecisionTreeClassifier(criterion='entropy', random_state=42)
```

In [11]:
```
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
                precision    recall  f1-score   support

           B       0.97      0.96      0.97       148
           M       0.93      0.95      0.94        80

    accuracy                           0.96       228
   macro avg       0.95      0.95      0.95       228
weighted avg       0.96      0.96      0.96       228
```

In [17]:
```
print(clf.predict([[
4,11.42,20.38,77.58,386.1,0.1425,0.2839,0.2414,0.1052,0.2597,0.09744,0.4956,1.156,3
```

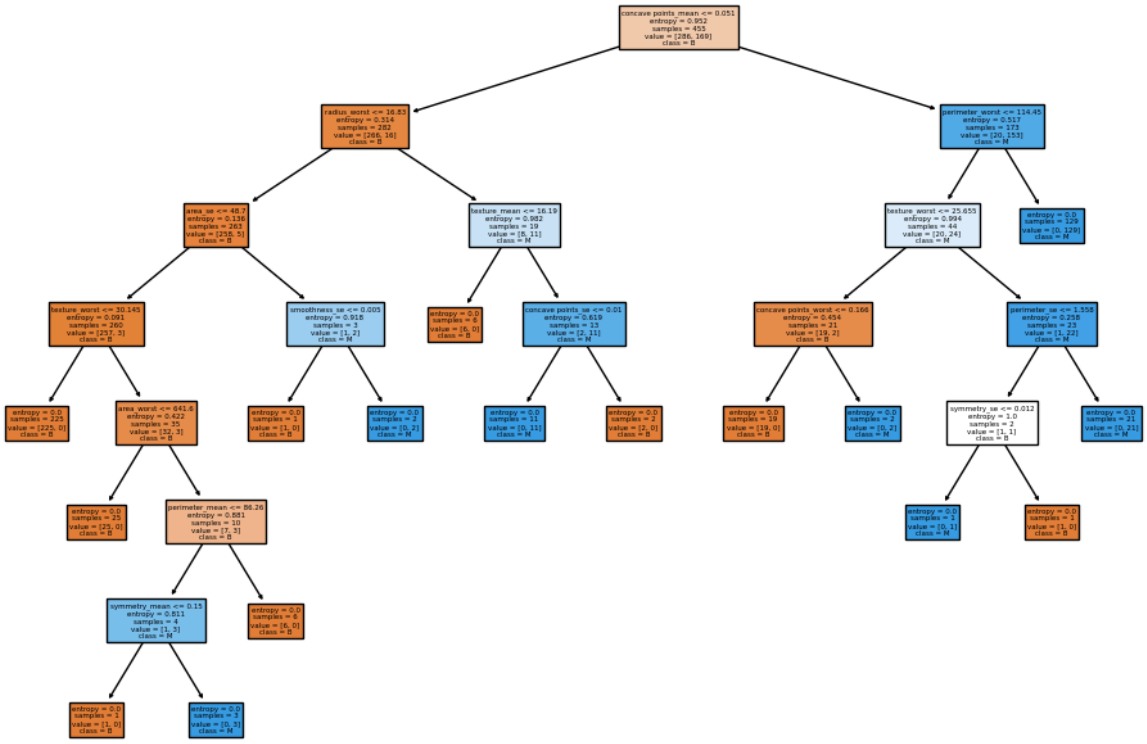
```
['M']
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does
not have valid feature names, but DecisionTreeClassifier was fitted with feature nam
es
  warnings.warn(
```

In [29]:
```
features_name=df.columns.drop(["diagnosis","id"])
print(features_name)
```

```
Index(['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
       'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```

In [23]:
```
plt.figure(figsize=(12,8))
plot_tree(clf, filled=True,feature_names=[
              'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smooth
              'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetr
              'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se'
              'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
              'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'sm
              'compactness_worst', 'concavity_worst', 'concave points_worst', 'symm
          ], class_names=clf.classes_)
plt.title("Decission Tree for fruit classification")
plt.show()
```

Decission Tree for fruit classification



In [ ]:

In [ ]:

In [ ]:

In [ ]: